Manuel de zenon

Réseau

v. 7.50

Table des matières

1. Bienvenue dans l'aide de COPA-DATA

AIDE GÉNÉRALE

Si vous ne trouvez pas certaines informations dans ce chapitre de l'aide, ou si vous souhaitez nous suggérer d'intégrer un complément d'information, veuillez nous contacter par e-mail : documentation@copadata.com (mailto:documentation@copadata.com).

ASSISTANCE PROJET

Si vous vous rendez compte que vous avez besoin de licences ou de modules supplémentaires, veuillezcontacter l'équipe commerciale par e-mail : support@copadata.com (mailto:support@copadata.com)

LICENCES ET MODULES

Si vous vous rendez compte que vous avez besoin de licences ou de modules supplémentaires, veuillezcontacter l'équipe commerciale par e-mail : E-mail   sales@copadata.com (mailto:sales@copadata.com).

2. Network

zenon networks can be set up quickly and securely.

zenon in the network allows you to, among other things:

- ▸ full access to the Runtime of different computers
  This way, actions such as the acknowledgment of alarms at a workspace on all other computers in the network thus becomes visible.

- ▸ Centralized logging and archiving

- ▶ Creation of redundant systems (see redundancy (à la page 77), circular redundancy (à la page 98))

- ▶ Redundancy switching with integrated evaluation methods

- ▶ Creation of distributed systems (see multi-project administration (à la page 30))

- ▶ Use of zenon Web Server and zenon Web Client for access using the web browser.

- ▶ Use of zenon in a terminal server environment (à la page 60)

- ▶ Use of strong encryption (à la page 47)

- ▶ Concurrent work on a project from several computers (see distributed engineering)

> **Informations concernant la licence**
>
> *Composante de la licence standard d'Editor et du Runtime.*

SIMPLE ADMINISTRATION OF THE ZENON NETWORK

The network functionality of zenon makes it possible to deploy projects in a distributed manner on different computers. You can thus create very efficient, complex network arrangements (à la page 10) with it. In doing so, arrangements can also be configured in such a way that certain project content is only visible at a certain location (a certain computer) for an activity. The zenon Editor supports users in creating and configuring such configurations.

The integrated topology administration (à la page 67) creates interrelationships for the individual projects in the process, with the attendant computers in graphical form. A testing routine checks the configured structure to see that it is complete and that there are no configuration errors.

With the network nodes function, zenon also checks to see if the selected network topology can function.

> **Informations**
>
> *With network projects, note the roles (à la page 9) (Primary Server, Standby Server or Client) in which modules and functions (à la page 130) can be administered and executed.*

WAN

zenon transfers data spontaneously with a network by default. With comprehensive project configurations, it is recommended that the Runtime monitoring between client and primary server is limited. The communication distance between the client and Primary Server can also be closed automatically. Routers only establish a connection if data exchange is absolutely necessary.

ZENON WEB SERVER

The zenon Web Server allows access to Runtime via the intranet or Internet. No adaptations to the project are necessary. Access is via the zenon Web Client. This offers the same functionality and display as zenon Runtime. The zenon Web Server is available as:

▶ zenon Web Server: Pure monitoring functionality.

▶ zenon Web Server Pro: Complete operation and monitoring functionality. It is possible to directly engage in processes over the web.

▶ **zenon Web Server Pro Light**: Limited functionality of zenon Web Server PRO for use with zenon Operator.

> **Informations**
>
> *You can find further information in the zenon Web Server manual.*

3. Roles of computers

With zenon, it is possible to create diverse network topologies. Starting with the simple client-server model through to comprehensive multi-hierarchical models.

IT-specific terms such as Server and Client are also used in zenon. However, in order to achieve unique identification of the individual components for complex multi-hierarchical structures with various computers and projects involved, we always speak of roles in zenon. Roles are always to be considered from the point of view of a project.

zenon Runtime can, on one computer, start one or more projects, depending on the project configuration (see also multi-project administration (à la page 30)). In doing so, the computer on which Runtime is started assumes one of the following roles for the respective projects:

▶ Primary Server

▶ Standby Server

▶ Client

These roles are shown below using examples in different topologies.

> **Info**
>
> If, in the course of this documentation, we speak of Primary Server, Standby Server or Client, the role that the computer takes for the project is always meant.

3.1   Terminology

The following terms are used for the role description in the zenon network:

Parameters	Description
Server:	Computer with connection to the PLC. The server takes on the administration of process and project data exclusively. Communication is checked by means of a watchdog (à la page 18). In the event of a server failure, the Standby Server undertakes its tasks, provided a standby was defined. As soon as the server is ready again, it automatically takes on its tasks and synchronizes all data.
Standby Server:	Takes on, in redundant systems, the role of the server, if this fails. It acts like a client in the network, but al saves all data like the server. In the event of hardware redundancy, the standby communicates with the redundant PLC both ways. The standby works with an internal buffer. Data loss during the downtime between server failure and the standby taking on the server role is thus avoided.
Clients:	Each computer on which Runtime is started is a client. Clients connect to the server to receive process data or to send this.
Data server:	The Data server is a computer with direct communication to the process but no server tasks. In normal operation, the data server sends all data to the Server, which manages data storage and data distribution. In the event of a network or server breakdown, the data server continues to work as a single station and therefore guarantees proper operation of the process. The data is transferred as soon as the server connection is re-established. Field of application: An on-site operating system without powerful hardware, such as an IPC or CE terminal. Complete data archiving and data distribution is carried out by the data server. On-site operation is still possible in the event of a server failure. This is not the case in a conventional Client-Server model.

💡 Informations
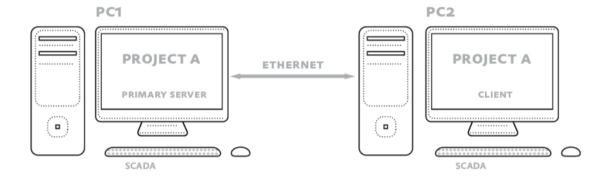
Server and client are not defined in relation to a computer, but in relation to a project.

If the names of the Server or Standby Server are changed, these cannot be reloaded. They are only updated by restarting Runtime.

3.2   Client-server model

In the Client-Server model, one computer is the Primary Server; all other computers are Clients.

▶   Computer 1 is the Primary Server for Project A.

▶   Computer 2 is the Client for Project A.

> 💡 **Informations**
>
> *You can find detailed information on this in the Multi-Project Administration (à la page 30) chapter.*

3.3   Redundant model

In the redundant model, one computer is the Primary Server and one computer is the Standby Server. All other computers are Clients.
If the Primary Server fails, the Standby Server takes on this role. All Clients connect to the new Primary Server.

REDUNDANCY WITHOUT CLIENTS

▶   Computer 1 is the Primary Server for Project A.

▶   Computer 2 is the Standby Server for Project A.

▶   If Computer 1 fails, Computer 2 is the new Primary Server for Project A.

REDUNDANCY WITH CLIENTS

▶   Computer 1 is the Primary Server for Project A.

▶   Computer 2 is the Standby Server for Project A.

▶   If Computer 1 fails, Computer 2 is the new Primary Server for Project A. All Clients automatically connect to Computer 2.

> 💡 **Informations**
>
> *You can find detailed information on this in the Redundancy (à la page 24) chapter.*

3.4   Multi-hierarchical models

It is possible to create different multi-hierarchical topologies with the help of multi-project administration. In doing so, it is possible to start several projects on one computer. The computer assumes a certain role for the project in the process.

> 💡 **Informations**
>
> Multi-hierarchical projects can also be executed without a network on standalone computers.

EXAMPLES

MULTI-CLIENT MODEL

▶   Computer 1 is the Primary Server for Project A.

▶   Computer 2 is the Primary Server for Project B.

▶   Project I runs on Computer 3 (integration project) as a standalone project and starts Projects A and B.

•   Computer 3 is a Client for both of these projects.

MULTI-SERVER MODEL

▶   Project I runs on Computer 1 (integration project) as a standalone project and starts Projects A and B.

•   Computer 1 is the Primary Server for both of these projects.

▶ Project I runs on Computer 2 (integration project) as a standalone project and starts Projects A and B.

- Computer 2 is a Client for both of these projects.

▶ Computer 3 is the Client for Project A.

▶ Computer 4 is the Client for Project A.

MULTI-CLIENT-MULTI-SERVER MODEL

▶ Project I runs on Computer 1 (integration project) as a standalone project and starts Projects A, B, C and D.

- Computer 1 is the Primary Server for projects A and B.

- Computer 1 is the Client for projects C and D.

▶ Project I runs on Computer 2 (integration project) as a standalone project and starts Projects A, B, C and D.

- Computer 2 is the Client for projects A and B.

- Computer 2 is the Primary Server for projects C and D.

▶ Project I runs on Computer 3 (integration project) as a standalone project and starts Projects A, B, C and D.

- Computer 3 is the Client for projects A, B, C and D.

> 💡 **Informations**
>
> *You can find detailed information on this in the Multi-Project Administration (à la page 30) chapter.*

4. Requirements

A basic requirement for using zenon is a functional Windows network.

GENERAL

The following prerequisites are necessary:

▶ TCP/IP as the network protocol

▶ Functional naming, can be chosen as **DNS**, **WINS** or local **HOST files**.

▶ Free TCP Port `1100`:
If a network project is loaded, zenon Runtime automatically starts the **zenNetSrv** network service. This service opens port `1100`. This must therefore be contactable remotely and must not be blocked by a firewall.

> 💡 **Info**
>
> zenon networks work with all supported operating systems.

IPV4 AND IPV6

The zenon network allows the choice of using IPv4 or IPv6. Dual operation is not possible. The setting is made via:

▶ in the **Network configuration** in the **Startup Tool**
or

▶ In zenon6.ini

If this setting is changed, all ongoing zenon processes must be restarted.
The services affected are:

▶ **zenAdminSrv**

▶ **zenSysSrv**

▶ **zenLogSrv**

▶ **zenDBSrv**.

The following components are not affected by the setting; they always use IPv4:

▷ Driver communication with the PLCs

▷ Protocol communication via process gateways

▷ Workbench and Runtime communication in zenon Logic

> ⚠ **Attention**
>
> *IPv6 only works with zenon version 7.00 onwards. If IPv6 is used, no versions prior to zenon version 7.00 can be started.*

PORTS USED

For communication within zenon, only TCP ports are used; no UDP ports are used.

The following ports are required for zenon in the network:

Service	File	Goal	TCP-port
Network service	**zenNetSrv.exe**	Runtime communication.	1100
Transport service	**zenSysSrv.exe**	Data transfer by means of Remote Transport (Editor).	1101
zenon Web Server	**zenWebSrv.exe**	On-site logging machine between web client and Runtime	1102

Port numbers can be amended individually by means of the **Listening ports** tab in the **Startup Tool**. Note in this case that all devices affected must be amended.

Informations

You can find further information in the Tools manual in the Startup Tool chapter.

STANDARD PORTS

Application	Port standard
zenon	
Network Service	1100
Transport Service	1101
WEB Service Classic	1102
DB Service	1103
SQL Browser Service (pour le développement distribué dans Editor)	1434
zenAdminSrv.exe	50777
Logging Service	50780
zenVNC.exe	5600 – 5610
SNMP Trap Service	50782
zenLicenseSr	50783
zenLicenseStub	50789
zenLicenseCenter	50689
WEB Service Tunneling	8080
zenon Logic	
Le port attribué à zenon Logic ou straton dépend du projet et du service. Par ex. : Le premier projet zenon Logic occupe les ports 1200 et 9000, le deuxième projet occupe les ports 1201 et 9001, et ainsi de suite.	1200 – 1210 4500 – 4510 7000 – 7010 9000 – 9010
zenon Analyzer	
Administration Service	50777
Analyzer Connector Service	50778
Analyzer License Service	50779
ZAMS	50781
Drivers	
Driver Simulation	6000 – 6020
Process Gateway OPC Server	135
Process Gateway SNMP	161
Process Gateway Modbus	502

Process Gateway IEC60870-5 104 slave	2402
Process Gateway DEC	5555
Process Gateway DNP3 Slave	20000

CHECK THE REQUIREMENTS

NAME RESOLUTION

To check the name resolution:

1. Start the windows command line
   (**cmd.exe**)

2. Enter the following command:
   ping [COMPUTER NAME]

3. If the name resolution is correct, you receive the IP address of the computer with Runtime as the answer; otherwise you receive an error message

TCP PORTS

To check the contactability of the TCP port 1100:

1. Start the zenon Runtime with a network project on a Remote computer:
   This starts the program **zenNetSrv.exe** and the TCP port `1100` is opened.

2. Start the windows command line
   (**cmd.exe**)

3. Enter the following command:
   `telnet RECHNERNAME 1100`

4. As soon as a connection is established, the content of the command line window disappears. Otherwise an error message will be displayed.

> ⚠ **Attention**
>
> The Telnet command is not part of the Windows operating system and must be installed separately. You can find instructions for this in the operating system help pages   (search for: `Telnet`).

4.1   Time synchronization in the network

With network projects, all computers in the network must be time-synchronized. zenon automatically carries out the synchronization necessary for this.

In a topology with several Primary Servers (such as circular redundancy (à la page 98)), it is recommended that time synchronization is implemented by means of an external time service (such as DCF77) or Windows resources. In this case, the automatic time synchronization in zenon must be deactivated.

> ⚠ **Attention**
>
> *If the time difference between the server and the client is more than 5 seconds, no more files are synchronized.*

AUTOMATIC TIME SYNCHRONIZATION IN ZENON

If the time synchronization is to be turned on or off manually, the following entry in **zenon6.ini** must be amended:

[Netz]

```
TIMESYNCH=1 -> automatic time synchronization active (default)
TIMESYNCH=0 -> automatic time synchronization inactive
```

EXTERNAL TIME SYNCHRONIZATION USING THE OPERATING SYSTEM

If the automatic time synchronization in zenon is deactivated, synchronization can be carried out via the operating system. To do this, a time server must be specified for this (with or without an external time service such as DCF77), which takes on the time synchronization with the other computers.

In the conventional Client-Server/Standby topology (without multi-project administration), the Primary Server is the active time master. This should keep its own time itself by means of an external time service if possible. The respective clients get the current time from this (depending on the timeout that has been set) and update their own times accordingly. Communication is via SNTP (`Simple Network Time Protokoll`). The delay in transfer is taken into account in the process.

💡 **Informations**

Watchdog

Time synchronization is carried out periodically at the set timeout time.

*When using the default setting of 30 seconds for the **Network communication timeout** property in the **Startup Tool**, the network service (**zenNetSrv.exe**) of each client sends a Watchdog to the network service (**zenNetSrv.exe**) of the Primary Server every 10 seconds during online operation. If the Primary Server responds to at least one of the three watchdogs within the 30 seconds, the client assumes that the network connection is working.*

Configuration in the Startup Tool:

Application -> Options-> Network configuration tab -> Option Network communication timeout.

Configuration in zenon6.ini:

Alternatively, the setting can be made directly in zenon6.ini:

[Netz]

NET_TIMEOUT_MSEC=*30000*

(timeout in milliseconds, default: 30000.)

Note the additional configuration necessary in WAN (à la page 20).

Attention: The minimum timeout time is 5 seconds. If a lower value is defined, this is interpreted as 5 seconds.

COMMANDS UNDER WINDOWS

For external synchronization using Windows, enter the following command with the respective necessary arguments in the console for command processing: **NET TIME [\\computer name | /DOMAIN[:domain name] : /RTSDOMAIN[:domain name]] [/SET] [/YES]**

Argument	Description
NET TIME	▸ Synchronizes the time of the computer with that of another computer or another domain or ▸ Displays the time for a computer or a domain If this command is executed without further arguments, then the current date and the current time of the computer that was defined as the time server for the domains is displayed.

Computer name	The name of the computers that checks or is to be synchronized.
DOMAIN[:Domain name]	The time is synchronized with the primary domain controller of the **Domain name** domain.
RTSDOMAIN[:Domain name]	The time of the computer is synchronized with a reliable time server from the **Domain name** domain.
/SET	Synchronizes the clock of the computer with the stated computer or the stated domain. After the command has been set, the server time is displayed and a request is made to see if this time is to be set.
/YES	Displays the current server time and synchronizes this with the local computer without a further request or confirmation.

Exemple

NET TIME \\Server /SET /YES

4.1.1   Time synchronization in the WAN

In a WAN and for dial-up connection, the standard defined value of 30 seconds for the timeout means that the connection would probably be maintained permanently.

Select a timeout time in the WAN that only initiates the establishment of a connection at the desired interval.

However, note: The longer the timeout, the later server failures are detected. For example, if you select `64800` as the time for the timeout, the timeout time is 18 hours. A connection is made every 6 hours and a watchdog is sent. A server failure would thus only be noticed after around 18 hours.

Informations

If no entry for the timeout is defined in **zenon6.ini***, the standard timeout of 30 seconds is used when Runtime is started.*

FUNCTION SCREEN SWITCH

Active data is requested when a screen is switched. Procedure:

▶ A check is made to see if a watchdog was sent to the Primary Server in the last 30 seconds.

▶ If this is not the case, a watchdog is sent to the Primary Server immediately. the waiting time for a response is 40 seconds.

▶ If a Server break down is recognized, the zenon network service automatically tries to reconnect every 30 seconds.

This would lead to a permanent connection establishment in the WAN network. This behavior can lead to entries in **zenon6.ini** being amended:

1. Open zenon6.ini.

2. Go to the **[NETZ]** section.

3. Create or edit the entry
   ```
   NET_CONNECTWAIT_MSEC=30000
   ```
   This defines the value for a reconnect in milliseconds.

   Maximum value: Timeout time

4. Create or edit the entry
   ```
   NET_CONNECTCOUNT=
   ```
   This defines the number of repetitions for a reconnect per cycle.

   The default is 0 repetitions, this means 1 attempt at reconnection.

5. Setting up the zenon network

TOPOLOGIES

zenon supports several network topologies:

▶ Client server network (à la page 23):
The same project runs on the server and all clients.

▶ Multi-server network:
A client can access different servers and thus display the data of different projects at the same time.

▶ Multi-client-multi-server model:
All clients and servers communicate with each other. Other projects can be accessed from each
project.

CONFIGURING THE NETWORK

To make a network network-compatible:

1. navigate to the **Réseau** node in properties

2. Activate the **Réseau actif** property

3. Use the **Serveur 1** property to define the computer that takes on the server role in the project
Note: The IP address is not sufficient; the name of the computer must be entered.

If necessary, you still configure the following in this section:

▶ Standby Server (à la page 94): **Serveur 2** property:

▶ Redundancy (à la page 77): **Type de redondance** property:

▶ **Message de fin**: Defines if, when Runtime is ended on a server, the clients are informed 70
seconds in advance

With this, you have configured the basic properties. Repeat these steps for all clients in the zenon
network. Remote computers can also be set up using remote transport.

> ⚠ **Attention**
>
> *Issue of a name for Serveur 1 and Serveur 2:*
>
> ▶ The computer name must be entered.
> The entry of an IP address is not permitted.
>
> ▶ `localhost` must not be used.
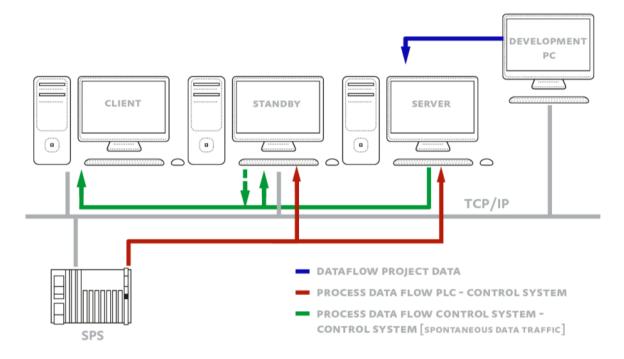
▶

5.1 Client-server model

In the conventional Client-Server model, only one project is used, which is started on all computers involved. In doing so, a certain computer is the Primary Server for this project. All other connected computers are Clients.

To set up the Client-Server model, the following must be the case in the project:

▸ The **Réseau actif** property must be activated

▸ The name of the computer that is to be the Primary Server is to be entered in the **Serveur 1** property

Recommendation: Select the most powerful computer in the network as the Primary Server.

In the zenon Client-Server model:

▸ Only the Primary Server has a direct connection to the PLC

▸ The Primary Server administers all process data (such as online data, archive data, alarms, recipes, etc.)

▸ The Primary Server administers all project data (such as screens, functions, defined variables, etc.)

▸ Each other computer starts as a Client in the network

▸ Each client establishes the connection to the Primary Server when Runtime is started,

• synchronizes the project data and

- displays the current process data

> 💡 **Informations**
>
> *The Client-Server model is fully supported under Windows CE. Windows CE devices can be used as Primary Servers or as a Client.*
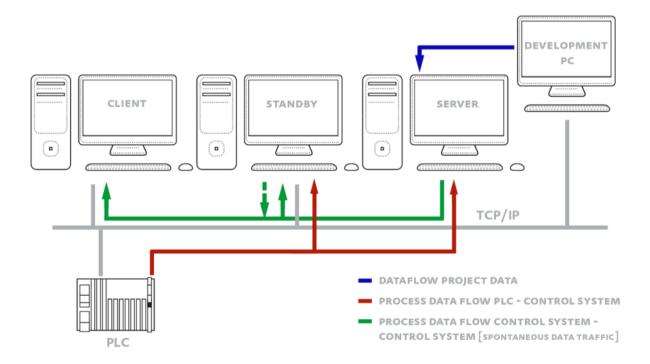
5.1.1   Redundancy

Redundant SCADA servers are used, if 100% process control and data safety is demanded, even when the server fails.

You achieve this fail safety by defining a second server, a so-called Standby Server, along with the project server. This standby automatically recognized a server failure If the Primary Server fails, the Standby Server takes on the complete functionality of the server.

In order to avoid data loss in the time between the server failure and the detection of the failure, the standby always buffers all data. This data buffering also takes place if the Standby Server is not the Primary Server. After a server failure this buffer is merged with the last data from the server and the new incoming data, so no data can be lost. The control system thus guarantees seamless redundancy.

Depending on the configuration of the **Mode de redondance** property, the original server is started either as a Server or as a Standby after restarting.

Redundancy mode	Description
Non-dominant	The original Server (**Serveur 1**) starts as a Standby. **Serveur 2** retains the
Dominant	The original Server (**Serveur 1**) takes on the role of Server again after all been synced. **Serveur 2** is downgraded to the Standby.
Rated	Depending on the configured rating (à la page 84), the original server is either as a Server or as a Standby after restarting.

Info

If the standby is running, when the server is started, the server copies all Runtime data from the standby. If you made any changes in the project data, while the server was offline and you have only updated them on the (not running) server, these changes will be overwritten, when the server copies the data from the standby.

In this case you have to update the standby before starting the server; or you close the standby when starting the server. As soon as it is restarted the standby then will get the new data from the (running) server.

CONFIGURING THE STANDBY

▶ In the network properties, enter, in addition to the **Server** (property: **Serveur 1**) the name of he **standby server** too (property: **Serveur 2**).

▶ To transfer the Runtime files, the process is the same as with the client.

The control system supports two different kinds of redundancy:

Software redundancy	The PLC is not redundant. Only the control system is redundant. (standard case)
Hardware redundancy	Both the PLC and the control system are redundant.

5.1.2 Configuring the server

The Primary Server makes the connection to the PLC and administers all data, both online data and configuration data. The Clients synchronize their data with the Primary Server.

To set up the Primary Server

1. Activate the **Réseau actif** property.

2. Define the computer that is to be the Primary Server for the project using the **Serveur 1** property.
    Note: The IP address must not be used; the name of the computer must be entered.

3. Note the correct configuration of the internal variable drivers (à la page 147)

4. Optional: Create AUTOSTART and AUTOEND scripts (à la page 147) for the clients if necessary

> ⚠ **Attention**
>
> *Issue of a name for Serveur 1 and Serveur 2:*
>
> ▸ The computer name must be entered.
>   The entry of an IP address is not permitted.
>
> ▸ `localhost` must not be used.

▸ If the development computer on which you created the project is also the Primary Server, configuration is now complete.

If the development computer is not the desired server, transfer your project configuration to the desired computer. The data can be transferred by means of Remote Transport or the network topology.

> 💡 **Informations**
>
> *You can find further information on this in the administer and check network topology (à la page 67) chapter or in the Remote Transport manual.*

5.1.3   Configuring the clients

Clients can be set up via Remote Transport, via the network topology or manually. Setup via Remote Transport or the network topology is recommended.

For this, the following applies:

▸ If the development computer is also is a client, start Runtime locally on that computer.

▸ You set up all other clients either by means of Remote Transport (à la page 27), network topology (à la page 67) or manually (à la page 28).

▸ If special processes are to be executed on the clients, a respective script in the project must be created, which defines the behavior on startup (AUTOSTART_CLIENT script (à la page 147)) and when being ended (AUTOEND_CLIENT script (à la page 147)).

> 💡 **Informations**
>
> *You can find further information on this in the administer and check network topology (à la page 67) chapter or in the Remote Transport manual.*

Set up client with Remote Transport

By default, Remote Transport always transports the Runtime files to the computer that is defined as the server in the **Réseau** project properties group. To set up clients from the development computer by Remote Transport, the Remote Transport connection must be set up before the Client is set up.

To set up clients using Remote Transport:

1. Open the **Général** node in Project Properties.

2. Click on the **Transport à distance (Remote Transport)** property.
   The dialog **Remote Transport** is opened.

3. Enter the name of the client in the network in **Connection** under **Name**.

4. Confirm the configuration by clicking on the **OK** button.

5. Establish a connection to Remote Transport.
   To continue to use Remote Transport, it is best to use the toolbar symbols.

6. Transport all Runtime files to the client with Remote Transport.

7. Set the start project for the client with Remote Transport.

8. Start the Runtime on the client with Remote Transport.

9. Close the Remote Transport connection

> 💡 **Informations**
>
> *Find out more information in the chapter Remote Transport.*

Setting up a Client by means of network topology

Network topology is suitable for setting up several Clients at the same time. In doing so, Runtime files can be transferred to several computers at the same time by means of multiple selection.

> 💡 **Informations**
>
> *You can find detailed information on this in the Network topology (à la page 67) chapter.*
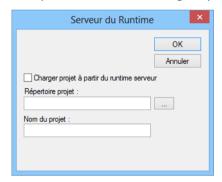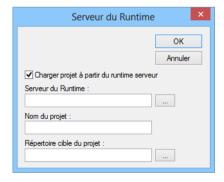
Setting up the client manually

To configure clients for the start of Runtime:

1. Close the zenon Editor and the zenon Runtime.

2. Open the file **zenon6.ini** with a text editor.
    You can find the file in the folder `%ProgramData%\COPA-DATA\System\`

3. Remove the line **VBF30 =....** or comment this line out.

    (Note: This entry defines which project is to be loaded when the Runtime is started.)

4. Leave the Editor closed and start Runtime.

5. A request is made in a dialog, requesting which project is to be loaded.

6. Activate the check box for the **Load project from Runtime server** option.

7. Configure the input field:

    a) **Runtime server**:
        Computer that is set up as the Primary Server (à la page 10). The name can be entered directly or selected from a list using the **...** button.
        You can find further information on configuration for a redundant network after these step-by-step instructions

    b) **Project name**:
        Name of the project that runs on the Primary Server.

> Note: Ensure that the project names are correct. If the folder name does not correspond to the project name, the project name must be changed here!

   c) **Project target folder**:
Folder for Runtime on the client's local hard drive. You either can select an existing folder using the **...** button or enter a path manually. If a folder that does not exist is entered, this is created automatically.

   d) Confirm the input by clicking **OK**.

8. These project configurations have the following effect in zenon Runtime.
Runtime:

   a) Creates a connection to the Primary Server

   b) Copies its Runtime files to a project target folder

   c) Starts Runtime

   d) Requires - if necessary - Runtime to be restarted

9. The entry **VBF30=...** in the `zenon6.ini` file is set on the project target folder.

Runtime then starts the network project automatically on the client each time it is started.
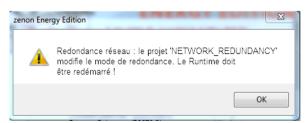
> ⚠ **Attention**
>
> Repeat this process for each client.

CONFIGURATION OF THE RUNTIME SERVER IN THE REDUNDANT NETWORK

With a redundant zenon network configuration, the issuing of roles for the primary server and standby server depends on the redundancy type. It is not always clearly recognized whether the configured **Serveur 1** or**Serveur 2** is in the role of the Primary Server.

You should therefore always enter both servers for the configuration in the **Runtime Server** dialog. The server names are separated from one another by a semi colon (;). Empty spaces for the server names are permitted.

Please note that the sequence of the entered servers corresponds to the sequence of the project configuration. The Runtime client does not start if the computer names do not correspond.

If the project configuration of **Serveur 1** or **Serveur 2** is changed, this change is discovered when reloading on the client. In Runtime, a dialog is displayed that informs you of the amended server configuration and forces the client to be restarted.

Ensure that the new Runtime files are transferred to the client again if necessary.

Behavior in Runtime

Network projects can be operated in the same way in the network by all computers in Runtime and Clients and are visualized in the same way. If there is no valid project defined when Runtime is started, the dialog to define the Runtime projects is opened. For details, see the Set up client manually (à la page 28) section.

Differences between the Primary Server and the Client:

▶ Only the Primary Server of the project has a connection to the hardware and administers the process data.

▶ The other computers (Clients) receive, from this:

• Current values of the variables

• Chronological Event List system messages

• Alarm information

• Recipes

• Archive data

• etc.

The transfer from the Primary Server to the Clients (such as a value change from a driver) is spontaneous and event-triggered (such as calling up a zenon `Extended Trend` screen, which needs archive data from a Primary Server).

MONITORING THE CONNECTION

When using the default setting of 30 seconds for the **Network communication timeout** property in the **Startup Tool**, the network service (**zenNetSrv.exe**) of each client sends a Watchdog to the network service (**zenNetSrv.exe**) of the Primary Server every 10 seconds during online operation. If the Primary Server responds to at least one of the three watchdogs within the 30 seconds, the client assumes that the network connection is working.

5.2   Multi-Project Administration

Multi-project administration makes decentralized solutions possible. Sub projects can be distributed to different computers. The individual computers in turn can be the Primary Server, Standby Server or Client for the respective sub projects.

The following is possible with the help of multi-project administration:

▶ Several projects in one workspace can be edited in the Editor at the same time

▶ Several projects can be started at the same time and thus variables, functions, archives etc. from other projects can be accessed directly throughout projects
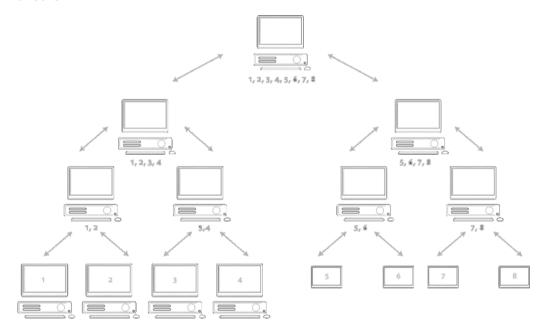
> 🔆 **Informations**
>
> *Multi-project administration is not available under zenon Operator. Here, only one project and a global project per workspace can be created and administered in the Editor. Runtime can only start one project.*

STRUCTURE

An integration project that is loaded in Runtime as a start project is required.

zenon creates a multi-hierarchical project tree, at the top of which is the integration project. Multi-project administration makes it possible to place the projects in a logical connection to one another.

> 🔆 **Informations**
>
> *Configure and check the topology with the zenon network topology (à la page 67).*

WORK EFFICIENTLY WITH MULTI-PROJECT ADMINISTRATION AND THE PROJECT HIERARCHY

zenon makes it possible to reuse data and screens from existing projects consistently. zenon multi-project administration makes a logical connection between the individual projects and places these in a hierarchical connection to one another. The user can display this project hierarchy graphically in the zenon Editor, by dragging the projects to the desired position with the mouse and thus creating a multi-hierarchical project tree.

The project that is highest in the hierarchy is the integration project. All other projects are subordinate to this project. The data from individual projects is available throughout all projects in the project structure.

The zenon multi-project structure is comparable to a file folder:

- ► Additional sheets – zenon projects – can be added at any time. The folder always automatically covers all information of the sheets stored in there. It is possible to browse through the pages at any time and look at the information, without taking the individual pages out. In the zenon multi-project structure, users can change between the individual screens or projects without having to take these out.

- ► The integration project can be compared to the contents of the file folder. It serves as a central navigation project and makes it possible to display screens or data from the subordinate projects. The individual projects are autonomous and can continue to be operated autonomously. Access from a project to the data or screens of another project is enabled via the zenon standard interfaces. Expansions or amendments to projects are made directly in the individual projects. Any maintenance work that may be carried out only has an influence on the respective project; the overall system remains unaffected by this.

MULTI-PROJECT ADMINISTRATION MEANS

- ► Small-sized, clear structures.

- ► Easy, quick and clear maintenance of the individual projects.
  It is possible, for example, to deactivate individual projects without influencing the others. In the same way projects can be distributed to different processors.

- ► Load distribution.

- ► Cross-project operation, as all projects on a processor are simultaneously activate.

- ► Multiple-hierarchy network structure allows the centralization of data (measured values, alarms, plant information, archive data, etc.) in a higher-ranking level.

- ► No limit on projects per processor.

- ► Summary of projects to large control rooms.

- ► Node structure – physical network separation.

5.2.1 Definition of the structure in the Editor

The structure is created by simply dragging & dropping in the Editor. You also need, in addition to productive projects, an integration project that administers all other projects. Because standalone projects do not send data to other computers, a Primary Server must be defined in each (sub-)project. The integration project itself can also be a network-capable productive project.
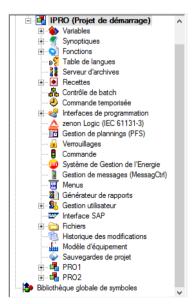
EXAMPLE

Three projects are used in this example:

▶ Integration project `IPRO`

▶ Productive project `PRO1`

▶ Productive project `PRO2`

To create the structure:

1. Create three projects:

   - `IPRO`

   - `PRO1`

   - `PRO2`

2. Define a Primary Server for each computer
   (The integration project can also be implemented as a standalone project)

3. In the Project Manager, drag `PRO1` to `IPRO` by holding the left mouse button

4. Do the same for `PRO2`

5. `PRO1` and `PRO2` are now displayed in the Project Manager as branches of the `IPRO`

The hierarchical structure of the network has thus been created.

> 💡 **Informations**
>
> *In order for elements of sub projects, such as screens, variables or functions to be able to be selected, the **Keep project in the memory** function (in the project context menu) must be activated.*

5.2.2   Transferring and starting projects

With the help of the network topology (à la page 67), the integration project can automatically be transferred to the respective target computer with all sub projects. All sub projects are also automatically transferred if the integration project is also the start project.

TRANSFERRING AND STARTING PROJECTS MANUALLY

Remote Transport can be used to transfer and start projects manually.
Each project is individually transferred to the corresponding computer.

- ▶ With Remote Transport, move all Runtime files of PRO1 to its Primary Server.

- ▶ Set the start project with Remote Transport.

- ▶ Start the Runtime with Remote Transport.

- ▶ Stop the online connection.

- ▶ Do the same for `PRO2`.

> 💡 **Info**
>
> In order to transfer the integration project and both sub projects to a Client via Remote Transport, 3 Remote Transport processes are necessary (a process for each project).
>
> Network topology (à la page 67) is preferred here, because all projects can also be transferred to several computers at the same time.

ADDITIONAL INFORMATION

- ▶ For details of network topology, read the Administering and checking network topology (à la page 67) chapter.

- ▶ You can find the configuration of the computer with an example for automatic transfer of sub projects in the Configuration of computers in the network (à la page 73) section.

5.2.3 Administering projects

You have the following possibilities, among others, for accessing the data from sub projects:

- ▶ Integration project (à la page 35)
- ▶ Navigation between projects (à la page 36)
- ▶ Using variables or functions from another project (à la page 37)
- ▶ Sending recipes to different variables in different projects (à la page 41)
- ▶ Create archives for use throughout projects (à la page 39)
- ▶ Creating a joint AML or CEL for different projects (à la page 43)

> ⚠ **Attention**
>
> *During configuration, note the roles in which the modules and functions are executed (Primary Server, Standby Server, Client). You can find a list of the possible configurations here: Behavior of modules in the network (à la page 130).*

The integration project

The integration project administers subprojects that can be accessed in Runtime. The integration project can be used as a pure administration project in multi-project administration (for example just for the navigation to the sub projects) or also as a complete productive project (with its own PLC connection, archiving, etc.). If the integration project is used as a start project, all sub projects are automatically started in Runtime.

In an integration project, you can for example create central Alarm Message Lists or Chronological Event lists for all subordinate projects with a few mouse clicks. This allows the alarms of all sub projects in the Alarm Message List of the integration project to be displayed in chronological order.

> ⚠ **Attention**
>
> *When designing the multi-project administration, ensure that the navigation (à la page 36) works.*

> ♀ **Informations**
>
> *These must be editable in order to be able to delete subprojects. For example, a subproject that has been created with an earlier editor version than the integration project can only be deleted after conversion.*

Navigation between projects

When administering more than one sub project in an integration project, it is absolutely necessary to ensure that it is possible to switch from one subproject to another or to the integration project in Runtime.
Hint: Create a template that is always in the foreground. Create a screen with navigation buttons based on this.

SCREEN SWITCH TO SUBPROJECTS

To switch between screens of individual projects, use the zenon screen switching function. In order for the navigation to be available at all times, first create a frame that is always in the foreground:
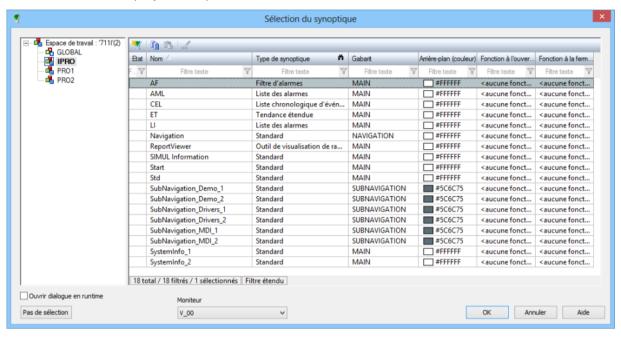
1. Create a new template that offers space for the navigation controls.

2. Assign it the **Toujours en avant plan** property.

3. Activate the properties **Type de cadre** and **Titre** (this means that the template can be moved in Runtime).
   Of course a template can also be used without a frame and title being in a fixed position.

4. Create a screen with navigation buttons on the basis of this template

EXAMPLE OF SWITCHING BETWEEN PRO1 AND PRO2

1. Create a new Screen switch function.

2. If there is more than one project available in the current workspace, the dialog to select a screen for the selection of a project is expanded.

3. Select `PRO1`.

4. Select the start screen of `PRO1` and close the dialog with **OK**.

5. Repeat points 1 to 4 for `PRO2`.

6. Add two text buttons with the text `PRO1` and `PRO2` to the navigation screen.

7. Link the two text buttons to each the functions that have been created.

> ⚠ **Attention**
>
> *zenon does not check in the Editor to see if the network structure in Runtime actually allows access to the selected project/screen.*
>
> *For example, in the Editor, screen switching to a screen in the integration project can be created in the project PRO1. This switching only works in Runtime if the integration project has also been started. This screen switching will not work on a computer on which the PRO1 project (start project) has been started.*
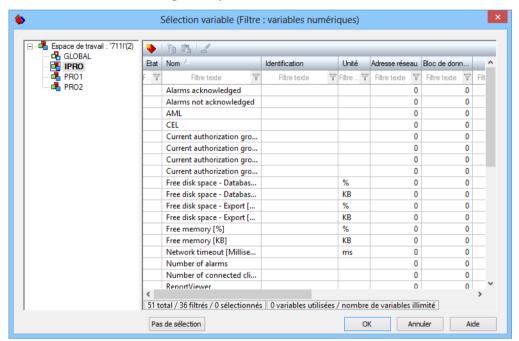
Variables and functions

You can access variables and functions from other projects in the same workspace directly by means of **dynamic elements**.

VARIABLE EXAMPLE

1. Open the `IPRO` start screen.

2. Add a new **counter value** dynamic element.

3. The variable selection dialog now opens.

4. Here, you can select not just variables from the `IPRO`. To select a variable from a different project:

   a) Click on a project in the left list area

   b) You are offered the attendant variables

   c) select the variable you want to assign

5. Select a variable from `PRO1` or `PRO2`.

Proceed in the same way for functions.

> ⚠ **Attention**
>
> *zenon does not check in the Editor to see if the network structure in Runtime actually allows access to the selected project and its variables/functions.*
>
> *For example, in the Editor, in project `PRO1`, a variable from the integration project can be selected. This connection only works in Runtime if the integration project has also been started. This connection will not work on a computer on which only project `PRO1` has been started (start project).*
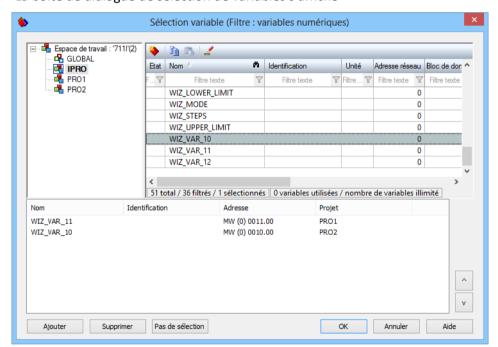
Archives

Les valeurs des variables de différents projets dans l'espace de travail peuvent être enregistrées dans l'archive. Les valeurs enregistrées de cette manière peuvent être filtrées, affichées sous forme de liste ou de tendance, et peuvent être imprimées ou exportées comme les données d'archives normales.
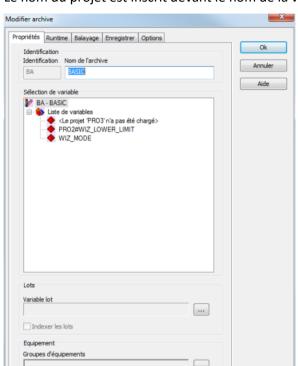
EXEMPLE D'ARCHIVE

1. Dans le projet `IPRO`, ouvrez le nœud **Serveur d'archives**.
2. Créez une nouvelle archive nommée `BA - BASIS`.
3. Ouvrez le menu contextuel de RECIPE1, puis sélectionnez **Ajouter variable**.
4. La boîte de dialogue de sélection de variables s'affiche

5. Ici, vous pouvez sélectionner des variables autres que celles du projet `IPRO`. Pour sélectionner des variables provenant d'autres projets :

   a) Cliquez sur un projet dans la liste affichée à gauche :
      Les variables associées sont affichées.

   b) Sélectionnez la variable que vous voulez attribuer.

6. Sélectionnez les variables parmi `PRO1` et `PRO2`.
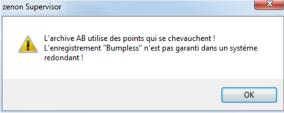
7. Le nom du projet est inscrit devant le nom de la variable la liste de variables de l'archive.

> ⚠ **Attention**
>
> *zenon n'effectue aucune vérification dans Editor pour s'assurer que la structure du réseau dans le Runtime permet d'accéder au projet sélectionné et à ses variables.*
>
> *Par exemple, dans Editor, dans le projet **PRO1**, une variable du projet d'intégration peut être sélectionnée. La connexion fonctionne uniquement dans le Runtime si le projet d'intégration a également été démarré. Cette connexion ne fonctionne pas sur un ordinateur sur lequel seul le projet **PRO1** a été démarré (projet de démarrage).*

Une fois la sélection des variables terminée, une boîte de dialogue d'avertissement indique qu'un enregistrement fiable est garanti dans toutes les circonstances.

EXEMPLE

▶ Le projet PRO1 est exécuté de manière redondante ; un ordinateur est le Serveur principal, et un deuxième ordinateur est le Serveur redondant.

▶ Les mêmes considérations s'appliquent au projet PRO2.

▶ Le projet d'intégration avec les projets sous-ordonnés PRO1 et PRO2 est démarré sur un troisième ordinateur. Il s'agit du Client pour tous les sous-projets.

Si les variables des projets PRO1 et PRO2 sont archivées dans le projet d'intégration, l'ordinateur reçoit des données concernant le réseau du Serveur principal correspondant de PRO1 et PRO2.
Par exemple, en cas de défaillance du Serveur principal de PRO1, pour la période de temps avant que le Serveur redondant de PRO1 ne reprenne le rôle de serveur, des valeurs de remplacement seraient fournies dans l'archive pour les variables issues de PRO1.

Remarque : Le buffer de redondance transparente du serveur de secours enregistre uniquement les variables du projet lorsque l'ordinateur a été configuré en tant qu'un des deux serveurs.
Solution : Pour garantir un enregistrement sans interruption, l'archivage doit être local dans un sous-projet exécuté de manière redondante.

Recipes
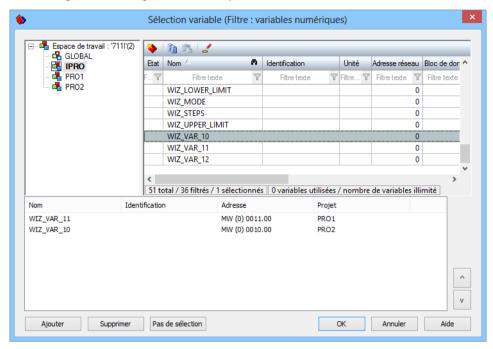
You can write values to variables from different projects of the workspace in a recipe.

RECIPE EXAMPLE

1. In the project IPRO open the node **Recipes**.

2. Create, under **Standard recipe recipes** a new recipe with the name **Recipe 1**.

3. Open the context menu of **Recipe 1** and select **Add variable**.

4. The dialog for selecting variables is opened

5. Here, you can select not just variables from the `IPRO`. To select variables from other projects:

   a) Click on a project in the left list area

   b) You are offered the attendant variables.

   c) Select variables from `PRO1` and `PRO2`.

   d) The name of the respective project is also placed in front of the variable name in variable list of the recipe.

Proceed in the same way for the Recipegroup Manager.

⚠ **Attention**

zenon does not check in the Editor to see if the network structure in Runtime actually allows access to the selected project and its variables.

*For example, in the Editor, in project **PRO1**, a variable from the integration project can be selected. This connection only works in Runtime if the integration project has also been started. This connection will not work on a computer on which only project **PRO1** has been started (start project).*
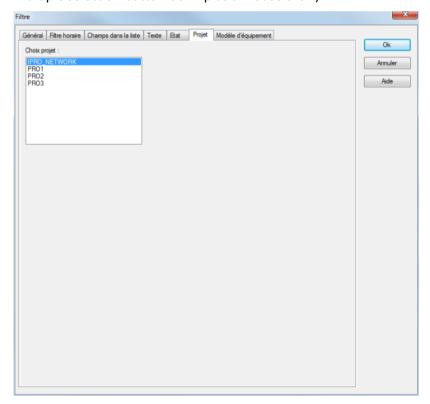
Alarms and CEL

In zenon, system messages and alarms from different projects of a workspace can be displayed together in a list. These entries can be filtered, displayed, printed or exported just the data from normal Alarm Message Lists or Chronological Event Lists.

AML example

1. Create an `AML` screen.

2. Add control elements to the screen via Control elements -> Add templates.

3. Create a function Screen switch for this screen.

4. The filter dialog for alarm lists is opened.

5. Open the **Project** tab.

6. Select the project that is to be displayed in the AML of the `IPRO`.
   Multiple selection button `Ctrl` plus a mouse click.)
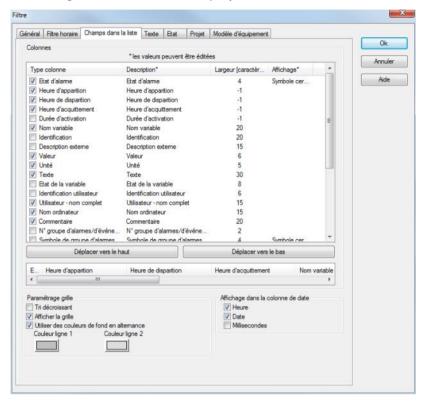
7. Open the **Column settings** tab.

8. Select the **Project name** property for display in Runtime.
   You thus gain an overview of the project from which an alarm comes in Runtime.
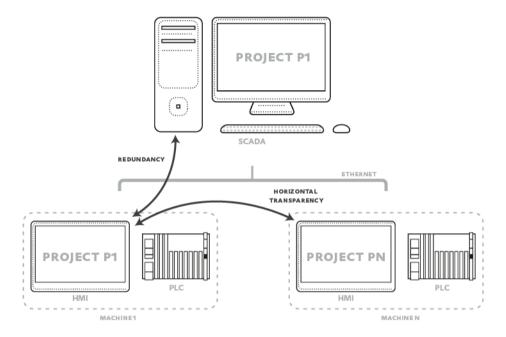
Proceed in the same way for the Chronological Event List.

5.3   Horizontal transparency

Multi-project administration also allows **horizontal transparency**.

Horizontal transparency means that all projects that are on the same level can be called up on one computer. The requirement for this is an integration project (à la page 35) with corresponding navigation, which starts these projects.

EXAMPLE

Several terminals belong to one machine. Each has its own visualization project. With the help of **horizontal transparency**, it is possible to show and operate, on each terminal, its own project and all other projects. This way the entire machine can be monitored and operated from each terminal.

5.4 Optimization for projects with a large number of Clients

Large network projects can, under certain circumstances with standard settings, place the Primary Server under full load for a period of time by reloading many Clients. The extent of the load depends on several factors (Primary Server resources, available bandwidth, etc.).
Guideline values:

▸ Runtime files of 10 MB or larger

▸ More than 50 clients

In this case, the reloading process can be optimized in order to prevent all Clients being reloaded at the same time. You undertake the project configuration of this in project.ini.

Possible INI entries for the optimization of the reloading process of clients in the zenon network:

▸ **RELOADDELAY_SEC**

▸ **CLIENT**

RANDOM RELOAD DELAY (RELOADDELAY_SEC)

With the **RELOADDELAY_SEC** ININ entry, the reloading is delayed by a random value.
To do this:

1. Open **project.ini** in the `\Projekt_SQL_Ordner\FILES\zenon\system\.` folder
    Hint: Highlight the project in the project manager and press the keys `Ctrl+Alt+E`; the
    Windows Explorer opens the `\Project_SQL_folder\FILES\` folder

2. Navigate to the section **[NETZ]**.

3. Create the entry **RELOADDELAY_SEC**=`[Value]`.

4. Select a `value` value for the delay.

    When reloading, a random delay in seconds is calculated for each client, which is between `0` and the selected value. `0` means no delay (standard action). The selected value has no influence on standalone projects, the Primary Server or the Standby Server.

    Note: This entry should only be set in every large projects with a noticeable delay when reloading. The standard settings provide better performance in normal projects.

DEFINED RELOAD DELAY (CLIENTX)

With the **CLIENTx** INI entry, the reload times for clients can be defined differently.

1. Open **project.ini** in the `\Projekt_SQL_Ordner\FILES\zenon\system\.` folder
    Hint: Highlight the project in the project manager and press the keys `Ctrl+Alt+E`; the
    Windows Explorer opens the `\Project_SQL_folder\FILES\` folder

2. Navigate to the section **[NETZ]**.

3. Create a **CLIENTx**=`[value]` entry for each client.
    x = the serial numbering for clients.
    `CLIENT[serial number]=client name,[reload delay in seconds]`

4. Select a `value` value for the delay.
    Example:

    *   `CLIENT0=VM-CDSBG104,5`

    *   `CLIENT1=WKS001,10`

Attention: the consecutive numbering of the clients must be continuously consecutive. If there are, for example, entries for CLIENT0, CLIENT1, CLIENT2, CLIENT3 and CLIENT5, only the entries for the following clients are taken into account: Client0, Client1, Client2 and Client3. The reload delay for Client5 is not taken into account.

> 💡 **Informations**
>
> For clients that do not have a **CLIENTx** entry, the random reload delay, as given in the **RELOADDELAY_SEC**entry, is applicable.
>
> If this entry is also empty, reloading takes place immediately.

6. Strong encryption of network communication

zenon enables strong encryption of communication in the zenon network. Strong encryption works from zenon Version 7.0 for all supported operating systems and for the zenon Web Client.

If encryption is active, communication between the Primary Server, Standby Server, Clients and zenon Web Clients is in encrypted form; the zenon Web Server only forwards data packets and is not affected by encryption.

> 🔆 **Informations**
>
> *Network communication was also encrypted in earlier versions of zenon. The method has changed with version 7. The term "encryption" in conjunction with zenon 7 or later always means strong encryption.*

6.1 Basics

Encryption for zenon Runtime is available from version 7.00. It is not possible to communicate with earlier versions of zenon if encryption is switched on. Encryption does not impair any zenon functionality.

BASIC ENCRYPTION FROM ZENON 7.00

To use the strong encryption of the zenon network, note:

▶ The password is encrypted individually on each computer and stored in **zenon6.ini**. That means:

- The password cannot be transferred by copying **zenon6.ini** to another computer.

- If hardware components are changed, in the network adapter area in particular, the password may be invalid and need to be re-entered.

▶ Encryption must always be activated or deactivated for all components involved in the zenon network. Communication between encrypted and unencrypted systems is not possible. **zenon Web Server**s only act as a proxy computer and are not affected by encryption.

▶ If encryption is activated on a computer, it always applies for the projects of this computer with the **Réseau actif** property active.

> ### 💡 Informations
>
> *AES 192 from Microsoft (https://msdn.microsoft.com/en-us/magazine/cc164055.aspx) is used as the encryption algorithm for network communication.*
>
> *SHA 256 from Microsoft (https://msdn.microsoft.com/en-us/library/system.security.cryptography.sha256%28v=vs.110%29.aspx) is used in order to generate the key from the entered password.*

COMPATIBILITY

Encryption is not compatible with versions prior to zenon 7.00 SP0. That means:

System 1	System 2	Communication
zenon 7 encrypted	zenon 7 encrypted	Yes
zenon 7 unencrypted	zenon 7 unencrypted or zenon prior to version 7 unencrypted	Yes
zenon 7 encrypted	zenon 7 unencrypted or zenon prior to version 7 unencrypted	No

Errors (à la page 53) are logged in the Diagnosis Viewer's LOG file.
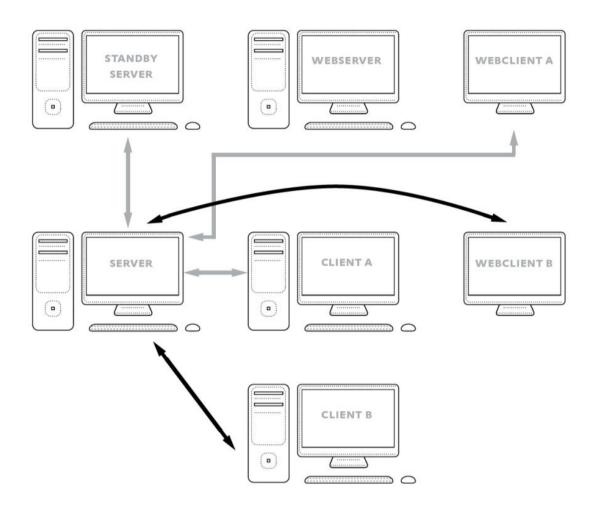
EXAMPLE

The following illustration shows an example of a network with Primary Server, Standby Server, two clients, one Web Server and two Web Clients. All devices are running zenon 7.00 SP0. The devices are configured as follows:

▶ Encryption is activated on the Primary Server using the Startup Tool (à la page 50).

▶ Encryption is also activated on the Standby Server and client A via Remote Transport (à la page 51) when Runtime files are transferred.

▶ Client B and Web Client B still communicate without encryption.

▶ On Web Client A, encryption is activated on the server using the Startup Tool (à la page 50).

▶ Because the Web Server does not evaluate the data packets, but instead forwards these on immediately, it does not require encryption. In theory, it can also have an older version, and the Web Clients can nevertheless create encrypted connections.

This configuration leads to the following result:

▶ The Standby Server communicates successfully with the Primary Server.

▶ Client A can log in to the Primary Server and exchange data.

▶ Because client B sends unencrypted messages and these are rejected by the Primary Server because encryption is active, client B cannot communicate with the Primary Server and is therefore offline.

▶ Web Client A logs on to the server via the Web Server and can exchange data.

▶ The unencrypted messages from Web Client B are forwarded from the Web Server to the Primary Server, but the server rejects these. Web Client B cannot communicate with the Primary Serverr and is therefore offline.

As soon as encryption via Remote Transport or the Startup Tool configuration on client B and via **Encrypt network communication** on Web Client B is activated, these connections can also make connections to the Primary Server.

6.2   Activate encryption

Encryption can be activated in different ways:

▶   via the Startup Tool (à la page 50) for the local computer and the Web Client

▶   via Remote Transport (à la page 51)

👍   **Conseil**

For quick, easy activation of the encryption, it is recommended that the configuration is carried our on a computer using Remote Transport (à la page 51).
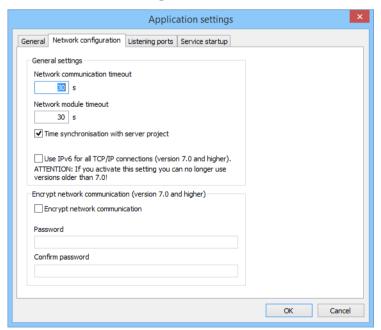
6.2.1   Locally via the Startup Tool

To activate encryption on the local computer or for the Web Client:

1.   Open the zenon **Startup Tool**.

2.   Click **Application -> Options**.

       The dialog for the settings is opened.

3.   Select the **Network configuration** tab.

4.   Activate the **Encrypt network communication.** checkbox

5. Enter the password and verify it.

6. Confirm the dialog by clicking on **OK**.
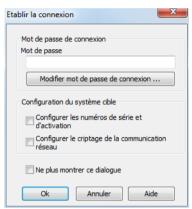
6.2.2 Via Remote Transport

Encryption can be activated on remote computers using Remote Transport. However, this is only possible if the Remote Transport connection is protected with a password.

To activate encryption using Remote Transport:
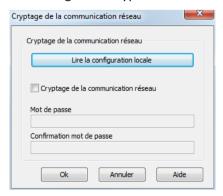
1. Click on the corresponding button in the Remote Transport toolbar

   or select, in the project's context menu: Set up Remote Transport> connection.

   The dialog for setting up the connection is opened

2. Enter the connection password or create one, if none has been set

3. Activate the **Configure encryption of network communication** checkbox

4. Click on **OK**.

   The dialog for encryption of network communication is opened

5. Activate the **Encrypt network communication** checkbox

6. Issue a password (for criteria, see the **network password encryption** (à la page 52) section.) To quickly transfer the local configuration to other computers, the local password can first be read via **Read local configuration**.

7. Confirm the dialog by clicking on the **OK** button.

6.3   Network encryption password

The following is applicable for encryption of the communication in the network:

▶ Minimum length `8 character`

▶ Maximum length: `20 characters`
  La longueur affichée est toujours définie sur `20 caractères`, afin de dissimuler la longueur réelle du mot de passe.

▶ Permitted characters:

  • Letters: `A – Z; a – z`

  • Digits: `0 – 9`

  • Special characters

▶ Permitted characters:

  • Space

  • `Enter` key (Return key)

▶ Summary: a password must contain at least 1 figure and 1 letter

6.4   Checklist for errors

In the event of errors, check:

▶ Do all computers have access to the network and does the naming resolution work between the computers?

▶ Was the **Réseau actif** property activated for the project in the Editor?

▶ Is zenon Runtime version 7.00 SP0 or higher being used? (relevant if encryption is being used)

▶ Is - for projects with encryption - the configuration correct on all computers? (**USE_ENCRYPTION** setting in **zenon6.ini**: The same for all computers, either `0`, `1` or not present.)

▶ Was the password set correctly?

▶ Was the hardware changed on one of the computers involved after the encryption has been configured?

▶ Does a ping work on a computer?

- Yes: Network connection present, fault is with the communication.

- No: Check the network.

▶ Is it possible to connect to Telnet?

- The connection is made: Both computers communicate at the same level. Check the password.

- The connection is made and lost again: One computer communicates with encryption, one without encryption.

- Faulty connection: zenon Runtime does not run on the target computer.

Note: Telnet must be installed as an extra on more recent Windows operating systems. Connection is generally made via port 1100.
The Telnet command is then: **open [IPAdresse] 1100**

▶ Are the required (operating system) functions available (primarily relevant for CE terminals)?

- Non-existent functions lead to Runtime not being able to start.

▶ If the service provider or one of the algorithms is not available, an error message (à la page 53) is written to the log file when Runtime is started.

Errors (à la page 53) are logged in the Diagnosis Viewer's log file.

6.5 Error message

Errors are either displayed in the output window of the zenon Editors, in pop-ups or in the LOG files of the Diagnosis Viewer.

NO CONNECTION

If a client has been configured with an incorrect encryption password (not the same as the password on the Primary Server) then this is evident from the following events:

▶ The Client is offline, although the Primary Server can be reached by pinging.

▶ The Primary Server writes error messages to the LOG file:

SysMod Error: Serialize in Object Project:[project name] Modul: [module number]

or:

NET Error During Decryption [Error number]

POP-UPS AND ERROR MESSAGES

Encryptions errors are notified by means of pop-ups (à la page 54) and entries in LOG files (à la page 57) or in the zenon output window (à la page 56).

6.5.1   Error messages in pop-ups

STARTUP TOOL AND WEB CLIENT

The following error messages are output by the zenon **Startup Tool** as a pop-up for local encryption or by the **Encrypt Network Communication Tool** for the configuration of the Web Clients.

These messages are always in English. Messages from Runtime itself are in the respective language that has been set.

Error message	Description
The password has to be entered in both text boxes!	When configuring the encryption, the user has left one of the two input fields (**Password** or **Password confirmation**) empty.
The passwords you typed do not match. Please retype the password in both boxes.	The content of the input field for password confirmation is different to the content of the input field for the password.
The network password does not fullfill the password criterias! Password criterias: - Minimum length = 8 - Maximum length = 20 - At least one character of the latin charset - At least one number - No spaces	The password entered does not meet the password criteria. The password criteria are displayed in the error message.
The network password could not be encrypted!	An error occurred when encrypting the network password.
The network encryption configuration in the file zenon6.ini is invalid. Please enter a new configuration.	When opening the **Network configuration** tab, it was established that the **zenon6.ini** file does not contain a valid configuration for the network encryption. A new configuration must be entered.
The network encryption password in zenon6.ini is invalid.	The password read in from the **zenon6.ini** file is invalid and must be reentered.
The password for network encryption is invalid and must be entered again!	Message when Runtime starts if the password cannot be verified.

REMOTE TRANSPORT

The following error messages are given by **Remote Transport** as a pop-up when the remote computer is encrypted.

Error message	Description
For configuring the network encryption, the Remote Transport connection must be protected with a password!	An attempt was made to configure remote encryption without securing the Remote Transport connection by means of a password.
You must enter the password in both input fields!	When configuring the encryption, the user has left one of the two input fields (**Password** or **Password confirmation**) empty.
The password confirmation does not match the password!	The content of the input field for password confirmation is different to the content of the input field for the password.
The password entered does not correspond to the password criteria. Password criteria: At least 8 characters Maximum 20 characters At least one letter At least one number No spaces	The password entered does not fulfill the password criteria. The password criteria are displayed in the error message.
An error occurred when encrypting the password!	An error occurred when encrypting the password. If this error occurs during configuration via Remote Transport, a more detailed error message is written to the log.
An error occurred when decrypting the network password from zeon6.ini.	The password stored in **zenon6.ini** cannot be decrypted. If this error occurs during configuration via Remote Transport, a more detailed error message is written to the log.
The encryption configuration in zenon6.ini is not valid and must be reentered.	The password read off from **zeon6.ini** is invalid. The password must be entered again.

6.5.2   Error messages in the output window

Errors are displayed in the output window as messages:

Message	Level	Description
`The server reports an error when compiling the data for the encryption configuration.`	**ERROR**	The remote **zenSysSrv** reports an error when compiling the information for the encryption of the password for network encryption. The adapter information cannot be read off.
`*** Configure the encryption of the network communication at the target system.`		This message is at the start of the conclusion message after encryption has been configured on a remote device via Remote Transport. After this, there is a message in relation to the success of the remote configuration.
`The server reports an error when the encryption configuration is changed.`	**ERROR**	The remote **zenSysSrv** reports an error when the encryption configuration is saved to the remote device. The configuration was not saved.
`The configuration was successfully saved on the server.`	**MESSAGE**	The remote **zenSysSrv** reports that the encryption configuration was successfully saved.
`The version of the remote zenSysSrv is too low. The encryption cannot be configured.`	**ERROR**	An attempt was made to configure the encryption on a remote device, which has a **zenSysSrv** from a version prior to version 7.00 SP0. Encryption is only available from zenon version 7.00 SP0; an earlier version of **zenSysSrv** cannot therefore configure this.

6.5.3 Error messages in LOG files

Errors in encrypted network traffic are documented in LOG entries. The **Error IDs** of the error messages in the following summary are system or COM error codes. You can find more information in the MSDN library.

LOG entry	Level	Description
`NET Error During Acquiring Cryptography Context [Error-ID]`	**ERRORS**	The creation of a service provider for encryption was unsuccessful.
`NET Error During Creating Hash [Error-ID]`	**ERRORS**	The creation of a hash value was unsuccessful.
`NET Error During Using Hash [Error-ID]`	**ERRORS**	The processing of a hash value was unsuccessful.
`NET Error During Destroying Hash [Error-ID]`	**ERRORS**	The release of a hash value that is no longer required was unsuccessful.
`NET Error During Deriving Key [Error-ID]`	**ERRORS**	The creation of a key for symmetrical encryption was unsuccessful.
`NET Error During Configuring Key [Error-ID]`	**ERRORS**	The setting of parameters for symmetrical encryption was unsuccessful.
`NET Error Cryptography Not Initialized!`	**ERRORS**	An encryption or decryption function was called up but initialization of the required parameters (service provider, key) was unsuccessful.
`NET Error Invalid Pointer passed!`	**ERRORS**	An encryption or decryption function was given invalid parameters.
`NET Error Message Length Must Not Be 0!`	**ERRORS**	The encryption function was called up with an empty message.
`NET Error During Buffer Length Calaulation [Error-ID]`	**ERRORS**	The calculation of required buffer size for encryption was unsuccessful.
`NET Error Buffer Length Must Not Be 0!`	**ERRORS**	The buffer for encryption or decryption has not been created.
`NET Error During Decryption 0x%x`	**ERRORS**	An error occurred during decryption.
`NET Error During Encryption 0x%x`	**ERRORS**	An error occurred during encryption.
`NET Error: Encryption Is Required And Project [Projekt] Received Plaintext Network Message`	**ERRORS**	Encryption is active and a decrypted message was received. The message is discarded in this case.
`NET Error: Encryption Is Not Supported And Project [Projekt] Received Encrypted Network Message`	**ERRORS**	Encryption is not active and an encrypted message was received. The message is discarded in this case.
`NET Cryptography Successfully Initialized`	**DEBUG**	The parameters required for encryption and decryption were initialized successfully. The parameters are initialized when Runtime is started.
`NET Uninitializing Cryptography`	**DEBUG**	The parameters required for encryption and decryption were released. This happens when Runtime

		is ended. If the log connection is separated before release, the message does not appear in the Diagnosis Viewer.
`NET Error During Buffer Size Calculation [Error ID]`	**ERRORS**	An error occurred when the necessary buffer size for compiling information for encrypting or decrypting the network password was calculated.
`NET Error During Buffer Size Calculation: No Adapters`	**ERRORS**	The computer does not have a network adapter. For this reason, the network password cannot be encrypted or decrypted.
`NET Error During Adapter Info Query [Error ID]`	**ERRORS**	An error occurred when the adapter information for encrypting or decrypting the network password was read off.
`NET Error Passwort Not Properly Formatted`	**ERRORS**	The hex dump of the encrypted password is in an invalid format.
`NET Error During Decrypting Password [Error ID]`	**ERRORS**	An error occurred when decrypting the network password.
`NET Error During Encrypting Password [Error ID]`	**ERRORS**	An error occurred when encrypting the network password.
`NET Cryptography Is Disabled`	**DEBUG**	Encryption of the network traffic is deactivated.
`NET Error No Password`	**ERRORS**	Encryption is active, but no password is entered.
`NET Error Password Could Not Be Decrypted`	**ERRORS**	The password for network encryption could not be decrypted.
`NET Password successfully loaded`	**DEBUG**	The password for network encryption has been loaded successfully.
`Network Cryptography Disabled By Remote Configuration`	**DEBUG**	**zenSysSrv** reports that encryption of network traffic on the computer was deactivated by the Remote Transport configuration.
`Network Cryptography Enabled By Remote Configuration`	**DEBUG**	**zenSysSrv** reports that encryption of network traffic on the computer was activated by the Remote Transport configuration.
`Network Cryptography Remote Configuration Error`	**ERRORS**	A configuration sent by Remote Transport for network encryption is erroneous.
`Error During Buffer Size Calculation [Error ID]`	**ERRORS**	An error occurred when the necessary buffer size for compiling information for encrypting or decrypting the network password for the configuration of Remote Transport was calculated.
`Error During Buffer Size Calculation: No Adapters`	**ERRORS**	The computer does not have a network adapter. For this reason, the network password cannot be encrypted or decrypted and thus not set via Remote Transport (it must therefore be connected via COM). The use of network encryption on a computer without

		a network adapter makes no sense however.
`Error During Adapter Info Query [Error ID]`	**ERRORS**	An error occurred when the adapter information for encrypting or decrypting the network password for configuration via Remote Transport was read off.
`NET Error During Decrypting Password: The Password is Invalid!`	**ERRORS**	The password is no longer valid, because the initial data for computer-dependent encryption has changed. This error can be rectified by configuring the password again. The decryption process is usually canceled before the validity of the password is checked, because the old password cannot be decrypted with the new encryption data. This leads to the „`NET Error During Decrypting Password 0x80090005`" error, which is usually displayed instead of „`NET Error During Decrypting Password: The Password is Invalid!`". Another consequence is that a password that is now invalid on the computer in question can lead to error messages when network packages are sent or received. The „`NET Error Cryptography Not Initialized!`" error message is written to the LOG file.

7. zenon on the Terminal Server

The zenon Runtime can also be used together with a terminal server solution.

Limitations:

▸ The Editor cannot run on a terminal server.

▸ Project simulation is not available for clients at the terminal server.

> 💡 **Informations**
>
> *Keep in mind that the name of the terminal client is resolved. That means: The name of the device that initiates the terminal connection is the name of the Clients for the zenon network.*
>
> *Attention: Ensure that all corresponding ports have been unlocked when using a firewall.*

Terminal server solutions are offered by several manufacturers. All tests with zenon were carried out using the Windows terminal server (Windows Remote Desktop Services).

> ⚠ **Attention**
>
> *When using zenon with a terminal server, it must be licensed with a Network dongle.*

PLUSIEURS INSTANCES DU RUNTIME

> *Une seule instance du Runtime de zenon peut être démarrée sur un ordinateur à un instant donné. Ceci s'applique, que le Runtime soit démarré en tant que fichier .exe, instance Web Client ou contrôle du Runtime (OCX).*

Exception : Sur le serveur de terminal ou le client du terminal, une instance de Runtime par utilisateur peut être démarrée en tant que fichier .exe, instance Web Client ou contrôle du Runtime (OCX). Une seule instance peut être exécutée à un instant donné dans un contexte d'utilisateur.

7.1   How terminal servers work

With terminal servers, it is possible to provide data and applications centrally, regardless of the end device. Terminal servers make it possible to start several shell instances (desktops) that are separate from one another on the terminal server.   If a client connects to the terminal server, a new shell instance is created and the client is assigned its own graphic user interface. The applications are executed on the terminal server itself and the data is saved on the terminal server.   Input (via the keyboard, mouse etc.) and output (display, audio, etc.) is on the Client.

> 💡 **Info**
>
> *Not all software is compatible with terminal servers.*

7.2   Advantages and disadvantages

The use of zenon offers the following advantages and disadvantages:

ADVANTAGES

- ▶ Only one computer (the terminal server) has to be maintained.
- ▶ Clients do not have to be very powerful (Thin Clients).

- ▶ Clients can have different operating systems (Windows, Windows CE, Linux, Unix, MacOS, iOS, Android etc.).

- ▶ High degree of data security, because no data is saved on the Client.

DISADVANTAGES

- ▶ All started programs of all instances run on one computer (the terminal server). This:

  - must have sufficient computing power for all started programs.

  - must have sufficient RAM for all started programs.

- ▶ All interfaces have to be shared. E.g. network adapters, COM ports, parallel ports.

- ▶ The network load can be high with an appropriate number of Clients (such as transfer of graphic data).

- ▶ The screen resolution is defined by the client started first. If screens in different resolutions are to be used, this can be implemented on the terminal server by means of an entry in **zenon6.ini** (**SERIALIZE**=1). All screens are then calculated again for the client, which further increases the necessary performance for the terminal server.

7.3   Operation zenon at the terminal server
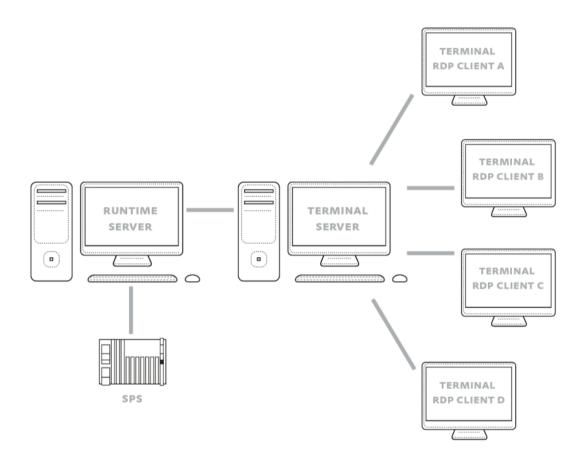
On a terminal server, operation is possible as:

- ▶ zenon Client possible without limitations

- ▶ zenon standalone systems only worthwhile and possible:

  - as a superordinate integration project to start several Client sub projects

  - without drivers, database connections etc.

- ▶ zenon Primary Server not possible

SCHEMATIC DISPLAY

Example topology of a terminal server network with zenon:

Computer	Description
Runtime Server	zenon Primary Runtime Server
Terminal Server	Terminal Server and n-fold Runtime Client
Terminal RDP Client a - D	Terminal Clients (input and output only)

7.4   Required settings

In order for zenon Runtime to be able to be started more than once on the terminal server, several settings must be made. The settings for registration, screen resolution and transfer service can be undertaken with the zenon Startup Tool.

💡 **Informations**

The following parameters are automatically set with registration via the Startup Tool:

▶ INI-entries:
   [TERMINAL]
   CLIENT=1
   SERIALIZE=1

▶ Registration of **ZenSysSrv.exe** as a service

▶ Deregistration of **ZenDBSrv.exe**

GENERAL SETTINGS

1. Registration

   Register the use on the terminal server via the zenon Startup Tool.
   Alternatively, you can configure the corresponding INI entries manually.

   zenon6.ini entry:
   The following entry must be added in zenon6.ini on the terminal server. On the Primary Runtime server no settings are needed.

   [TERMINAL]
   CLIENT=1

   `1`: The Runtime can be started several times, all settings for the terminal server operation are automatically set by the Runtime.
   `0`: The Runtime can only be started once. Operation on the terminal server is not possible. (standard setting)

2. Automatic adjustment of the screen resolution

   Per default the first client at the terminal server defines the screen resolution. This setting can be amended on the terminal server with the following entry in zenon6.ini:

   [TERMINAL]
   SERIALIZE=1

   `1`: Each client gets its own resolution.
   `0`: The first client defines the resolution.

3. Transfer

The transport service (**ZenSysSrv.exe**) must be registered and started as a Windows service, not as a standard EXE file. This setting is automatically set when registering via the Startup Tool, but can also be set manually.

Manual setting:

a)   Start the program from the command line interface with the **-service** option.

For example: **C:\Programs (x86)\COPA-DATA\zenon750\zenSysSrv.exe -service**

b)   Then start the Windows Service Manager. The service will be started automatically during every computer restart.

Note: The setup always registers the transport service as a standard EXE. Therefore the transport service must be re-registered as a Windows service after every reinstallation.

4.   Runtime folder

All users must have write access to the Runtime folder. All Windows users (Windows users: **All**) in Windows Explorer must have complete access to the Runtime folder and all its subfolders.

⚠   **Attention**

If TERMINAL=1 is set, the project simulation is no longer available.

SELECTIVE RELOADING OF SINGLE PROJECTS

Projects can also be synchronized selectively. In this case clients only reload projects if project changes exist. To activate the selective reloading:

1.   open the file **zenon6.ini** with a text editor

2.   go to section **[TERMINAL]**

3.   edit or create entry: **CLIENT_NO_FILE_ALIGN=**

4.   possible values:

0: Projects are always reloaded by all clients

1: selective synchronization active. Only the zenon client which is started in the console session of the terminal server synchronizes the Runtime files with the zenon server

After synchronizing the Runtime files, the console client writes the file **reloadindicator.tmp** to the directory that contains **project.ini** file of the program . The session clients at the terminal server check every 10 seconds whether this file is available. Does it exist and is its file time stamp newer that the date of the last reload, a session client reloads automatically.

ENTRY IN ZENON6.INI FOR SELECTIVE RELOAD

[TERMINAL]

CLIENT=1

CLIENT_NO_FILE_ALIGN=1

SERIALIZE=1

> 💡 **Informations**
>
> *You can get further information configuration files manual, in the Terminal server [TERMINAL] chapter.*

7.5   zenon Remote Desktop versus Terminal Server

The terminal server solution is different from zenon Remote Desktop primarily due to the following points:

zenon Remote Desktop connection

All connected stations always see one and the same desktop. If e.g. one user starts a program, all see the same program, the same mouse cursor, the same keyboard input, etc.

Terminal server connection

Each connected station has its own desktop - an own instance. Only it sees, what happens there. Mouse actions and keyboard inputs only affect this one instance.

That also means:

In each instance a program can be started separate, e.g. a text editor. The program then runs on the terminal server several times and therefore needs more resources.
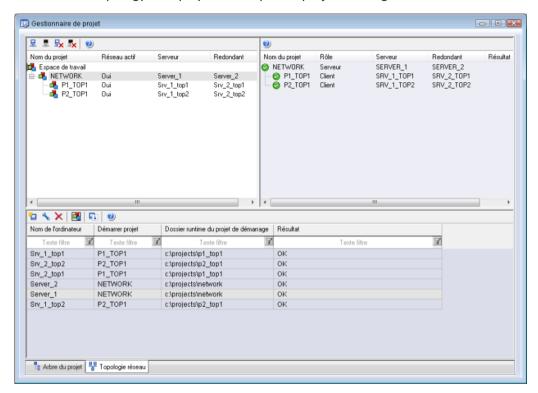
> 💡 **Informations**
>
> You can also find further information in the zenon Remote Desktop manual

8. Administering and checking network topology

The network topology is displayed in a separate project manager tab.
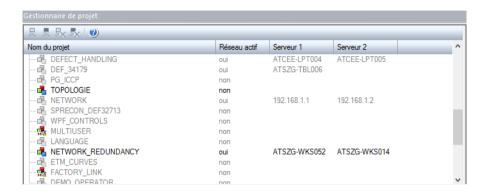
The display is divided into three areas:

▶ Topology tree (à la page 67) (top left):
  shows active projects; the global project is not displayed

▶ Event tree (à la page 69) (top right):
  only the result is displayed; represents the topology tree of a selected computer

▶ Computer list (à la page 70)(bottom):
  List display and configuration of computers in the network

8.1 Topology tree

The topology tree displays active projects in hierarchical form.

Parameters	Description
Project name	Is defined in the project tree tab and cannot be changed here.
Network active	Displays whether the network option is active for this project. The setting can be changed via the **Réseau actif** property. ▸ `Yes` project is a network project ▸ `No` project is not a network project
Server 1	Displays the Primary Server defined for this project. The setting can be changed via the context menu, the symbol in the toolbar or the **Serveur 1** property.
Server 2	Displays the Standby Server defined for this project. The setting can be changed via the context menu, the symbol in the toolbar or the **Serveur 2** property.

8.1.1   Tool bar and context menu
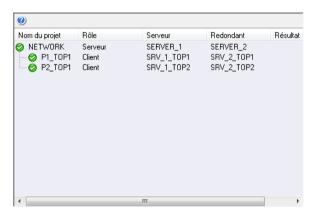
Property	Description
Set computer as Primary Server	Defines the computer highlighted in the computer list (à la page 70) as the Primary Server for the project highlighted in the tree.
Set computer as Standby Server	Defines the computer highlighted in the computer list (à la page 70) as the Standby Server for the project highlighted in the tree.
Delete Primary Server	Deletes the Primary Server defined for the highlighted project. Note: Configured computer is removed from the **Serveur 1** property.
Delete Standby Server	Deletes the Standby Server defined for the highlighted project. Note: Configured computer is removed from the **Serveur 2** property.
Help (? symbol)	Ouvre l'aide en ligne.

8.2   Result tree

The result tree represents the project tree of the computer selected in the computer list (à la page 70) from the project, which is set as a start project for the selected computer and displays these project settings.

The result tree is empty if:

 ▶ The start project of the selected computer was not found

 ▶ More than one computer in the list was selected
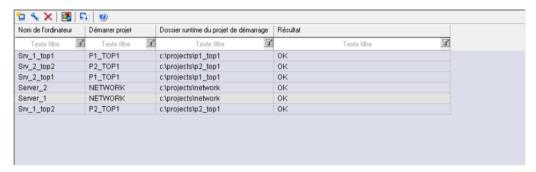
Parameters	Description
Project name	Projects that are assigned to the selected computer.
Role	Role of the computer: ▸ Primary Server ▸ Standby Server ▸ Client
Primary Server	Name of the computer that acts as a Primary Server to Runtime.
Standby Server	Name of the computer that acts as a Standby Server to Runtime.
Result of test	Shows detailed error messages (à la page 75) for topology test.

8.3 Computer list

The computer list shows computers that have been set up in the project and allows them to be configured. The list relates to the workspace and is saved in the workspace file (***.wsp6**).

The cells can be sorted and filtered. The column width can be amended with a right mouse click.

Nom de l'ordinateur	Démarrer projet	Dossier runtime du projet de démarrage	Résultat
Texte filtre	Texte filtre	Texte filtre	Texte filtre
Srv_1_top1	P1_TOP1	c:\projects\p1_top1	OK
Srv_2_top2	P2_TOP1	c:\projects\p2_top1	OK
Srv_2_top1	P1_TOP1	c:\projects\p1_top1	OK
Server_2	NETWORK	c:\projects\network	OK
Server_1	NETWORK	c:\projects\network	OK
Srv_1_top2	P2_TOP1	c:\projects\p2_top1	OK

Parameters	Description
Computer name	Name of the computer. Can be changed by: ▸ Click in the cell: Clicking on the **…** button opens a drop-down list of the computers currently available in the network. Note: This is only possible for pre-existing entries. ▸ **Edit computer** entry in the context menu or tool bar. Opens the dialog to configure the computer (à la page 73) in the network. ▸ **Nom de l'ordinateur** property.
Start project	The start project assigned to the computer Can be changed by: ▸ Click in the cell: Select from drop-down list. Note: This is only possible for pre-existing entries. ▸ **Set start project** entry in the context menu or the toolbar Sets the project selected in the topology tree (à la page 67) as the start project. ▸ **Projet de démarrage** property.
Start project Runtime folder	Folder for project files on the target computer. The files of the start project are saved in this folder. All other projects relating to this correspond to the structure of the **Dossier du Runtime** set up on the local computer. For example: **Dossier Runtime du projet de démarrage**: `C:\Projecs\Top  =` location where start project is stored. Sub projects are stored in `C:\Projects\`. Hint: Use the project name as folder name in order to create the same structure as on the engineering computer automatically. The **Dossier Runtime du projet de démarrage** can be changed by: ▸ double clicking on the computer: Opens the Configure computer dialog (à la page 73) in the network. ▸ Click in the cell: Manual entry possible. ▸ **Dossier Runtime du projet de démarrage** property.
Result of test	Shows the result of the topology test. ▸ **OK:** All projects are free of errors. ▸ **Error detected - For details see detail view!** One or more projects have an error. ▸ **Critical error detected - For details see the detail view!** A project has a serious error. Serious errors halt further testing.

	▶ **Not tested, because there is a serious error in the structure**: The computer was not fully tested, because the test was ended due to a serious error.
	Detailed error messages (à la page 75) are displayed in the result tree.

8.3.1   Tool bar and context menu
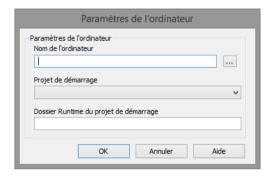
Entry	Description
Add computer...	Opens the Configure computer dialog (à la page 73) in the network.
Edit computer...	Opens the dialog to configure the computer (à la page 73) in the network for these computers.
Delete computer	Deletes computer from the topology after requesting confirmation. Multiple selection is possible. Attention: Deletion of the Primary Server or Standby Server leads to serious errors in the topology.
Set start project	Sets the project selected in the topology tree (à la page 67) as the start project.
Copy Runtime files from all projects on the computer	Copies all projects valid for the selected computer to the target computer. The result is displayed in the output window.
Help (? symbol)	Ouvre l'aide en ligne.

8.3.2   Computer network configuration dialog

The following configurations are necessary to configure the computers:

Parameters	Description
Computer name	Clicking on the **...** button opens the dialog to select a computer from the list of computers that are currently available in the network. Note: If many computers are available in the network, it can take some time to call up the dialog.
Start project	Selection of the start project from a drop-down list. All projects that are currently available in the workspace are displayed.
Start project Runtime folder:	Folder for project files on the target computer. The files of the start project are saved in this folder. All other projects relating to this correspond to the structure of the **Dossier du Runtime** set up on the local computer. Hint: Use the project name as folder name in order to create the same structure as on the engineering computer automatically. For example: Project name = **I-Project** at **Dossier Runtime du projet de démarrage** enter: `C:\Projects\I_Project` The sub projects in relation to this path are stored at `C:\Projects\Projektname`, for example: The project name is **SubProject1**, then the Runtime folder for this is `C:\Projects\SubProject1`. Requirement: The Runtime folders are left at their default settings and the projects were created at one file level. If this is not the case, it may be the case that subprojects cannot be copied, because the relative folder cannot be created from the start project. Example: The integration project has the following set up as a Runtime folder: `C:\Workspace\Projects\I_Project`. The sub project has the following set up as a Runtime folder: `C:\Subproject`. The start project Runtime folder is set to `C:\Project`. The sub project cannot be transferred, because the relative folder would be `..\..\Project`. This does not work, because the Runtime folder for the sub project would be below `C:\`. Solution: Set the **Dossier du Runtime** project property correctly. It is best to do it so that the Runtime folder is at the same level for all projects.

FERMER BOÎTE DE DIALOGUE

Paramètres	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

8.4 Error messages from topological testing

The topology test is always carried out if settings concerning the topology change. The effect of each change can be observed immediately this way. The topology is also tested if the topological view is changed.

TESTS CARRIED OUT

- ▶ Is the project defined in the project tree available in the project tree?

- ▶ Was a Primary Server defined?

- ▶ Were different computers defined for Primary Server and Standby Server?

- ▶ Can the client achieve its Primary Server/Standby Server?

- ▶ Can the Primary Server reach its clients?

- ▶ Can the Standby Server reach its clients?

- ▶ Is the Primary Server available for a project in the topology?

- ▶ Is the Standby Server available for a project in the topology?

- ▶ Is a computer included more than once in the path from Client to the Primary Server?

NOT TESTED:

- ▶ Is a client only updated on one path by the Primary Server or do several paths exist?

CLIENT TO SERVER

- ▶ Does the client reach its Primary Server via the Primary Server's chain?

- ▶ Was a computer that routes switched to its Standby Server?

  Info: The server must also be able to be reached by the client via the project's Standby Server that routes.

ERROR MESSAGE

Errors that are recognized during the topology test are displayed in the result tree (à la page 69) in the test result column.

Error	Reason	Solution
`The start project is unknown!`	Start project cannot be found.	Correct project configuration or include missing project in the workspace.
`The computer is entered as a server and standby!`	Server and Standby Server must be different computers.	Define different computers as Server and Standby Server.
`No computer is entered as server!`	The project is a network project but no server was configured.	Define a computer as a server.
`The project is not started on computer [name]! However it is necessary because higher hierarchic levels need access to it.`	The project is not loaded on the computer stated. The project is however routed via this.	Adapt topology or start project for the computer or deactivate the **Routage actif** property.
`Circular access to the server: The computer [name] redirects to the client [name]!`	The routing path from the client to server goes around in a circle. The computer that acts as a node redirects to the client.	Adapt topology or start project for the computer or deactivate the **Routage actif** property.
`Circular access to the standby: The computer [name] redirects to the client [name]!`	The routing path from the client to the standby goes around in a circle. The computer that acts as a node redirects to the client.	Adapt topology or start project for the computer or deactivate the **Routage actif** property.
`The computer [name] is not included in the list of computers`	Computer is missing in the list of computers for the topology.	Add computer to topology.
`Not checked because there is critical error in the topology`	No check could be made out for this computer because there was a critical error.	Rectify other errors so that the check can also check this computer.
`Circular access path: [computer names]`	The routing path from the client its the server/standby goes around in a circle. The "computer names" field contains the names of the computer that are affected. Structure: The first computer is always the client. The separator between the computer names indicates whether the following computer is a server or standby. ▸ > identifies the following	Adapt topology or start project for the computer or deactivate the **Routage actif** property.

	computer as a server. ▸ "+" labels the following computer as a standby. For example: Computer1+Computer2 > Computer3	
`Circular access path exists from server to client!`	The server is found according to the node when searching for the client computer.	Adapt topology or start project for the computer or deactivate the **Routage actif** property.
`The server (name) cannot reach this client!`	The path is not closed from the stated server to the client. The client is included in the client list on the server but is not updated. (The blue dots do not disappear on the client.)	Adapt topology or start project for the computer or deactivate the **Routage actif** property.

9. Redundancy

Redundancy in zenon ensures that processes are not interrupted even in the event of a failure of the Primary Server and that no data is lost. In zenon, we speak of **seamless redundancy**.

Seamless redundancy means that the time period between the failure of the Primary Server and detection of the failure is protected from data loss. This is implemented as follows:

▸ The Standby Server detects the failure of the Primary Server and automatically takes on the complete functionality thereof.

▸ In order to avoid data loss in the time period between the server failure and the detection of the failure, the Standby Server always buffers all data.

▸ After a failure of the Primary Server, this buffer is merged with the last data from the server and the new incoming data, so no data can be lost.

Info

*For **zenon Operator**, depending on the role, the following is applicable for connections to the server with Supervisor license:*

Client

▶ Connection to **Serveur 1** and **Serveur 2** is possible.

▶ Client connects to the Standby Server if the Primary Server fails.

▶ Several instances of a driver are supported.

▶ All variables of all drivers are displayed and can be written.

▶ Archives can be read.

▶ Redundancy switching can be executed.

Standby Server:

▶ Start as a Standby Server is possible.

▶ A warning message must be confirmed.

▶ Variables of the Standby Server are displayed.

▶ Archives of the Standby Server are displayed.

▶ Only the first instance of a driver is started on the Standby Server.

▶ After redundancy switching, variables of the second instance of a driver have the status `INVALID`. This status also remains after being switched again.

Data server:

▶ More than one instance of a driver can be started. However only one of the instances can send values to Runtime. zenon Operator thus behaves along the lines of a CE panel on which only one driver can be started for technical reasons.

Standalone

▶ A warning message must be confirmed.

▶ Variables of the Standby Server are not displayed.

Informations

Project changes need only be entered on the Primary Server; the standby server and the connected clients automatically synchronize online data. This ensures that the project status is the same on all computers.

SOFTWARE REDUNDANCY AND HARDWARE REDUNDANCY

With redundant systems in zenon, a distinction is made between software redundancy (à la page 79) and hardware redundancy (à la page 79).

CIRCULAR REDUNDANCY

zenon circular redundancy (à la page 98) is a special form of redundancy that allows seamless redundancy for several projects with a low amount of hardware being used and very simple configuration.

ZENON REDUNDANCY AND ZENON LOGIC REDUNDANCY

You can find details on the combination of zenon redundancy and zenon Logic redundancy in the zenon Logic Runtime manual).

> 🔆 **Info**
>
> *If only one controller is available, which offers only one communication channel, the **Stop at the Standby Server** option in the general settings of the driver configuration can be activated. The driver is thus stopped at the Standby Server and only started again at the upgrade.*

9.1   Types of redundancy

For redundant systems with zenon, a distinction is made between:

▶ **Software redundancy:** (à la page 80)

- Only the Primary Server communicates with the controller in two directions (write and read).
- The Standby Server communication is read-only. In addition, the standby server buffers the data in order to take on the role of the Primary Server in the event that this fails, without any loss of data.

▶ **Hardware redundancy:** (à la page 80)

- PLCs are designed as redundant.
- The Primary Server communicates with the controller by writing and reading.
- The Standby Server communicates with the second controller by writing and reading.

> 🔆 **Informations**
>
> *You can find further information of the configuration of the Standby Servers in the Setting up the Standby Server (à la page 94) chapter.*
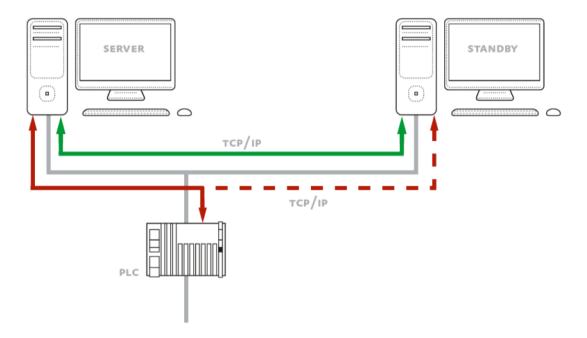
9.1.1   Software redundancy

The following components are required for software redundancy:

- ▶ A controller
- ▶ Two redundant computers (Primary Server and Standby Server)

In operation, the Primary Server communicates with the controller in two ways, the Standby Server communicates with the controller as read-only and synchronizes its data from the Primary Server.

In the event of a failure of the Primary Server:

- ▶ The Standby Server becomes the new Primary Server
- ▶ Seamless redundancy (à la page 92) ensures that all data is complete without omissions, including data from the time between the failure and the switch
- ▶ The new Primary Server communicates with the PLC both ways

9.1.2   Hardware redundancy

In zenon networks, there are two types of hardware redundancy available:

- ▶ Dominant and non-dominant (à la page 81)

- In a non-dominant network, the Primary Server can communicate with the first controller in two directions.

- The Standby Server communicates with the second controller in two directions.

- The Standby Server synchronizes with the Primary Server.

- The second controller synchronizes with the first controller.

▶ Evaluated (à la page 82)
  In a rated network, configurable criteria are evaluated in a matrix. The server roles are given according to the result.
  Attention: Please note the limitations in rated networks if a CE terminal acts as a data server.
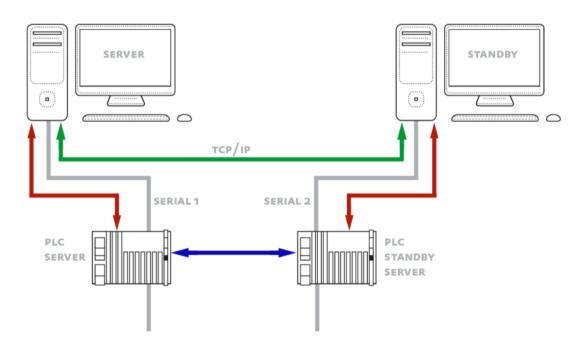
9.1.3   Hardware redundancy

In a non-dominant and dominant network, the Primary Server communicates both ways with the first controller; the Standby Server communicates with the second controller both ways.
The Standby Server synchronizes with the Primary Server; the second PLC synchronizes with the first PLC.
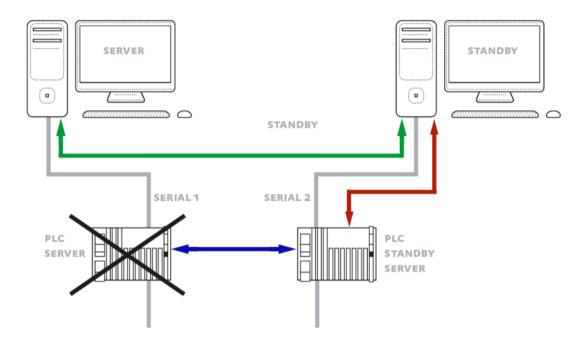
In the event of a failure of the Primary Server:

▶ The Standby Server becomes the new Primary Server

▸ The Seamless redundancy (à la page 92) ensures that all data is complete without omissions, including data from the time between the failure and the switch

▸ The Primary Server communicates both ways with the second controller

9.1.4 Redundancy in a rated network

With hardware redundancy in a rated network, evaluation criteria (à la page 84) decide which computer takes on the role of the Primary Server and which takes on the role of the Standby Server.

This evaluation can be freely configured and can include several different criteria. Each criterion is assigned evaluation points - the weighting. The sum of the weighting points then decides on the respective server role.

Informations

If one of the two servers - regardless of whichever role they have - loses the other server (for example: hardware failure), it automatically upgrades to the Primary Server.

REASONS FOR DELAYING THE REGRADING:

▶ The internal modules have not yet been fully initialized. This is possible if a driver delays the regrading process.

▶ The regrading is already active and not yet completed.

▶ The file sync is still active.

▶ Reloading is taking place.

Note: As a result of the automatic cyclical, renewed activation of the regrading, this is therefore only suspended until the process is permitted.

⚠ **Attention**

If, in a rated network, server redundancy switching is triggered due to an evaluation, there is no guarantee that pending functions in the queue of the old Primary Server have also actually been processed successfully before switching.

This is because during redundancy switching in rated networks, both servers take on the role of the Primary Server for a short time. In this time, Serveur 1 and Serveur 2 synchronize their data. The time period of this status depends on the network load and can be between 200 and 500 milliseconds.

TIME SYNCHRONIZATION

In a rated network, the time synchronization of **Serveur 1** and **Serveur 2** should always be deactivated by means of the zenon network.

The following is recommended for this:

▶ Deactivate the automatic time synchronization in zenon.

▶ Activate external time synchronization on **Serveur 1** and **Serveur 2** by means of the operating system

💡 **Informations**

You can find detailed information on this in the Time synchronization in the network (à la page 18) chapter.

Redundancy in a rated network

The following properties of the **Réseau** properties group must be configured for the configuration of an evaluated network:

1. **Réseau actif** must be `activated`.

2. **Serveur 1** and **Serveur 2**

3. **Type de redondance** is `software redundancy`

4. **Mode de redondance** is `evaluated`

5. Redundancy evaluation in the **Évaluations** property field.
    You can find details on evaluation under "Configuration of redundancy evaluation (à la page 84)".

6. **Délai de commutation [s]**

7. **Temps mort après commutation [s]**

8. **Hystérésis** (the evaluation points)
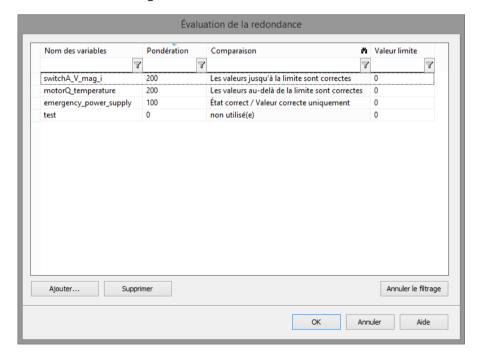
Configuration of redundancy evaluation

The dialog to configure redundancy evaluation is opened by clicking on "**...**" in the **Évaluations** properties field in the **Réseau** properties group.

The result of this evaluation is evaluated with the system variables **[network] Evaluation result for Server 1** and **[Network] Evaluation result for Server 2**.
Note: You can find further information on the network system variables in the System driver manual, in the Topic - network chapter.

The fields in this dialog can be filtered and sorted.

Nom des variables	Pondération	Comparaison	Valeur limite
switchA_V_mag_i	200	Les valeurs jusqu'à la limite sont correctes	0
motorQ_temperature	200	Les valeurs au-delà de la limite sont correctes	0
emergency_power_supply	100	État correct / Valeur correcte uniquement	0
test	0	non utilisé(e)	0

Évaluation de la redondance — Ajouter... — Supprimer — Annuler le filtrage — OK — Annuler — Aide

Parameters	Description
Name of the variable	List of the variables that can be used for the evaluation. Note: A maximum of 200 variables can be used for the redundancy evaluation.
Weighting	Value of the test criteria. Provides the value of the individual test criteria in the entirety of the test results. The weightings, of all variables that are true are counted up. Input range: `0 – 1000` Default: `100`
Comparison	States the type of comparison, from which the value of the weighting is taken into account in the overall assessment. Drop-down list: ▸ `Not used` No comparison to the weighting is carried out. ▸ `Only status OK /value OK` Only a check to see if there is a value is carried out. As soon as there is a valid value, the value of the weighting is used for the overall evaluation. ▸ `Values from limit are OK` Values that are greater than or the same as the limit that has been entered use the value of the weighting in the overall evaluation. ▸ `Values up to limit are OK` Values that are less than or the same as the limit that has been entered use the value of the weighting in the overall evaluation. Default: `not used`
Limit	Limit for comparison with "`Values from limit are OK`" or "`Values up to limit are OK`" Default: `0` Note: If the comparison is "`not used`" or "`Only status OK /value OK`", entries in this field have no influence on the evaluation.
Add...	Opens the dialog to select variables.

Remove	Removes the selected variable from the list Note: Multiple selection is possible.
Cancel filtering	Removes the filter criteria

Paramètres	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Annuler	Annule toutes les modifications et ferme la boîte de dialogue.
Aide	Ouvre l'aide en ligne.

Exemple

A variable is configured in the evaluation dialog with the following settings:

▸ Weighting: `50`

▸ Comparison: `Values from limit are OK`

▸ Limit: `50`

Set the value of the variable to 55. The condition becomes true as a result and the weighting of 50 is added to the result of the system variable **[Network] Evaluation result for Server 1**.

a further variable is configured with the following settings:

▸ Weighting: `30`

▸ Comparison: `Values up to limit are OK`

▸ Limit: `10`

As long as the value of this second variable =< 10, the weighting is added to the system variable.

If the value of the second variable is 5 for example, the **[Network] Evaluation result for Server 1** *(if the previously-described variable is true) is 80.*

Switching delay, down time and hysteresis

SWITCHING DELAY

The **Délai de commutation [s]** is the time (tolerance time) in which value changes of the evaluation result do not trigger a regrading of the server. In doing so, short peaks or a brief failure do not lead to an immediate server change.

Note: A further improvement of the evaluation result during the configured switching delay does not reset the timer.

> 📧 **Exemple**
>
> If a **Délai de commutation [s]** of 30 seconds is configured, the evaluation on the current Standby Server must be at least 30 seconds better than that of the Primary Server in order to trigger regrading.

After successful regrading, the timers of **Temps mort après commutation [s]** and **Délai de commutation [s]** switching delay can run at the same time; both must have expired in order to trigger a regrading.

DOWN TIME

This setting prevents resetting on the basis of the Primary server during the configured time.

Note: the "Redundancy switching (à la page 126)" function can be executed.

> 📧 **Exemple**
>
> If the evaluation of the current Standby Server - after the last regrading within the set down time - is higher than that of the current server, there is not another regrading until this time expires. The start of the "time calculation" is the time in which the role swap of the server is completed.
>
> If the evaluation increases within the down time of the Standby to more than that of the Server, but it goes down again to below or the same as the server before it expires, there is no regrading.

HYSTERESIS (IN EVALUATION POINTS)

The configuration of the **Hystérésis** with evaluation points. If a **Hystérésis** is configured, a check is made before redundancy switching to see whether the evaluation results between the two computers (**Serveur 1** and **Serveur 2**) correspond to the configured evaluation points. The switching is triggered once the value has been reached.

> 📧 **Exemple**
>
> **Serveur 1** is Primary Server.
>
> **Serveur 1** and **Serveur 2** each have evaluation result `0`.
>
> The **Hystérésis** is configured with value `100`.
>
> If the evaluation changes from **Serveur 2** to `99`, nothing happens; if the value reaches `100`, a regrading is triggered.

Switching delay - example

In this example, the switching of the two servers is triggered by a higher evaluation of the **Serveur 2**.

Once the reason for switching has occurred, the delay time is waited before **Serveur 2** becomes the Primary Server.

Although **Serveur 1** could take on the role of the Primary Server again, the delay time is waited for before **Serveur 1** takes on the role of the Primary Server.

As can be seen in the screen, **Serveur 2** takes on the role of the Primary Server once the difference of the evaluation system variables is higher than the hysteresis of the delay time. The delay time is counted after switching.

Serveur 2 takes on the role of the Standby Server at the time when the difference is higher than the hysteresis again, once the switching delay has expired.

Note: The role exchange of the two servers is visualized in the graphics by the respective vertical dashed line.

LOG entries in the rated network

The following log entries are written in the Diagnosis Viewer for the rated network:

LOG entry	Debug Level	Description
`LOG SendData Project:RATED_NET To:CDSBG079.COPA-DATA.INTERNAL Modul:10 Prior:0 Class:CD_CSystemDaten`	**Debug**	Data from the CD_CSystemDaten class is transported via connection 0 (watchdog) to computer `cdsbg079`. The module gives the zenon module to the one that has the data. `Module 10:` Synchronous execution network module
`LOG SendData Project:RATED_NET To:W7X64 Modul:10 Prior:0 Class:CD_CSrv1SystemDaten`	**Debug**	As above, the target is `Server2` and the source is `Server1`
`LOG ReceiveData Project:RATED_NET To:S Modul:10 Class:CD_CSrv2SystemDaten`	**Debug**	As above, the target is `Server` and the source is `Server1`
`<Beschreibung> Prj:<ProjektName> client:<ClientRechner> Cmd:<CommandTxt>(<CommandNum>) Timeout:<Timeout>, ReqId:<ReqId> reload:<Reload> stat1:<SubCommand> stat2:<Stat2> string:<AddText> list:<ItemCnt>`		Description of the function for which the message is created. e.g.: Server received system data.; Client send watchdog; etc. `ClientRechner`: Source or target depending on direction `CommandTxt`: Command as plain text. e.g.: `Server_REQ_DateiListe` `CommandNum`: Numerical identification of the command: ▸ 0 = Server_REQ_LifeMsg, ▸ 1 = Server_REQ_DateiListe, ▸ 2 = Server_REQ_GetDatei, ▸ 3 = Client_REQ_UpdateProjekt, ▸ 4 = Server_REQ_RedundanzUmschaltung, ▸ 5 = NETSRV_ConnectionClosed, ▸ `Timeout`: Timeout in ms `ReqId`: Request ID `Reload`: 1 if Server is in Reload, otherwise 0 `SubCommand`: Number of subcommand. Meaning depending on CommandNum For Command 0 ServerReqLifeMsg: ▸ 1 = STAT_CLIENT_ABGEWIESEN, ▸ 2 = STAT_CLIENT_ANGENOMMEN

		▸ 3 = STAT_CLIENT_SERVERCLOSE
		▸ 4 = STAT_CLIENT_SERVERSWITCH
		▸ 5 = STAT_CLIENT_SB_ANGENOMMEN
		▸ 6 = STAT_SB_SERVER_TO_SB
		▸ 7 = STAT_PEER_ALIVE
		For Command 4 redundancy switching:
		▸ 0 = IdleChangeOfDominance
		▸ 1 = AdviseChangeOfDominance
		▸ 2 = ConfirmChangeOfDominance
		`Stat2`: Additional command-specific information. e.g.: With command Server_Req_LifeMsg, subcommand: STAT_CLIENT_ANGENOMMEN, response for a Client logging on to a Server is the project version number in HiWORD.
		AddText: Additional text.
		ItemCnt: Number of objects in the list.

`Client send watchdog.`
`Prj:RATED_NET`
`client:ATSZG-WKS053.COPA-DATA.INT`
`ERNAL Cmd:Server_REQ_LifeMsg(0)`
`Timeout:30000, ReqId:0 reload:0`
`stat1:2(STAT_CLIENT_ANGENOMMEN)`
`stat2:1 string: list:0`
`quantifier:<Quant>`
`role:<RoleTxt>(<RoleNum>)`
		Message from **Serveur 1** to Server2
		
		`Quant`: Current evaluation.
		
		`RoleTxt`: Current role as text
		
		`RoleNum`: Current role as number

`Server received system data.`
`Prj:RATED_NET client:WKS053-W7X64`
`Cmd:Server_REQ_LifeMsg(0)`
`Timeout:30000, ReqId:0 reload:0`
`stat1:2(STAT_CLIENT_ANGENOMMEN)`
`stat2:1 string: list:0`
`expSrv1role:<ExpRoleTxt>(<ExpRole`
`Num>)`
		`ExpRoleTxt`: Role expected by Server2 for **Serveur 1** as text.
		
		`ExpRoleNum`: Expected role as number

`Manual role switch for project %s` `denied.`	**WARNINGS**	Manual redundancy switching cannot be carried out because a regrading is already active.

`Current evaluated server1 role:`
`%s(%i) Performing manual role`
`switch.`
	DEBUG	A manual redundancy switch is carried out. **Serveur 1** must switch its role according to the message.

`Unknown standby logon!` `Project:<PrjName> new client <Pc>!`	**ERROR**	Message is logged if a Client logs on as a Standby Server, but the Client name does not correspond to that of **Serveur 1** or **Serveur 2**. The identification is carried out through the host name. `PrjName`: Project that logs on to the Standby Server. `PC`: Name with which the Standby Server logs on.
`Unknown standby logon!` `Project:<PrjName> new client <Pc>!`	**ERROR**	Message is logged if a Client logs on as a Standby Server, but the Client name does not correspond to that of **Serveur 1** or **Serveur 2**. The identification is carried out through the host name. `PrjName`: Project that logs on to the Standby Server. `PC`: Name with which the Standby Server logs on.
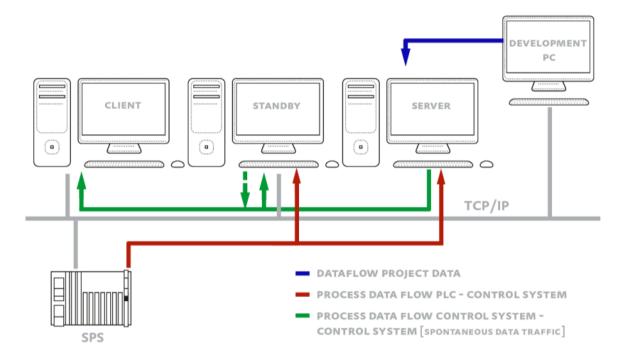
9.1.5   Configuration of seamless redundancy

For the configuration of seamless redundancy for a zenon project, you need two (conventional) computers.

Define, in the project properties:

1. A computer as Primary Server (à la page 25) for the project
2. A computer that is to be the Standby Server (à la page 94) for the project

PROCEDURE FOR SEAMLESS REDUNDANCY

- ▸ One computer is the Primary Server and one computer is the Standby Server for a project.
- ▸ As in a normal Client-Server model, the Primary Server has ownership of all data.
- ▸ The Standby Server acts outwardly to the user like any other computer on the network that has started the project.
- ▸ The Standby Server independently records all historical data such as alarms, events and archives and synchronizes other data (recipes, users, etc.) with the Primary Server.
- ▸ If the Primary Server fails, the Standby Server upgrades itself and takes over its tasks.
  - • In order to avoid data loss in the time period between the server failure and the detection of the failure, the standby always buffers all data.

After a failure of the Primary Server, this buffer is merged with the last data from the server and the new incoming data, so no data can be lost.

▶ All Clients connect to the new Primary Server.

▶ If the original Primary Server goes online again, a distinction must be made between when redundancy mode is used:

DOMINANT

▶ The original Primary Server goes online again.
  In doing so:

- It connects to the current Primary Server.

- It synchronizes all data.

- It becomes the Primary Server itself again.

▶ The original Standby Server becomes the Standby Server again.

▶ All Clients connect to the new Primary Server.

NON-DOMINANT

▶ The original Primary Server goes online again.
  In doing so:

- It connects to the current Primary Server.

- It synchronizes all data.

- It becomes the Standby Server itself.

▶ The current Primary Server remains the Primary Server.

▶ All Clients retain the connection to this.

RATED

▶ With evaluated redundancy mode, the roles of Primary Server and Standby Server are given on the basis of an evaluation matrix.

▶ Both computers each carry out an evaluation calculation on the basis of configured evaluation criteria.
  The current Primary Server is then the computer that has the higher evaluation.

▶ If both computers have the same evaluation, no exchange of roles is carried out.

▶ Alarms and CEL entries are each written by the Primary Server.

▶ The Clients each connect to the Primary Server.

> 💡 **Info**
>
> *Project changes need only be entered on the Primary Server; the standby server and the connected clients automatically synchronize online data. This ensures that the project status is the same on all computers.*

SOFTWARE REDUNDANCY AND HARDWARE REDUNDANCY

With redundant systems in zenon, a distinction is made between software redundancy (à la page 79) and hardware redundancy (à la page 79).

CIRCULAR REDUNDANCY

zenon circular redundancy (à la page 98) is a special form of redundancy that allows seamless redundancy for several projects with a low amount of hardware being used and very simple configuration.

ZENON REDUNDANCY AND ZENON LOGIC REDUNDANCY

You can find details on the combination of zenon redundancy and zenon Logic redundancy in the zenon Logic Runtime manual).

> 💡 **Info**
>
> *If only one controller is available, which offers only one communication channel, the **Stop at the Standby Server** option in the general settings of the driver configuration can be activated. The driver is thus stopped at the Standby Server and only started again at the upgrade.*

9.2 Setting up the Standby Server

To set up the Standby Server:

1. In the project properties in the zenon Editor, switch to the **Réseau** group.

2. Enter, in the **Serveur 2** property, the name of the computer that is to serve as the Standby Server for the project.
   Note: the computer must have a connection to the PLC.

   You can enter the computer name:

a) By selecting from the drop-down list by clicking on the **...** button
The **Select the desired computer** dialog is opened with all computers available in the network.
Note: This process can take a very long time, depending on the number of computers available in the network.

b) By entering the server name in the input field

3. Select, in the **Type de redondance** property, the desired type of redundancy (à la page 79) from the drop-down list:

> 💡 **Informations**
>
> *Servers from different domains are permitted!*
> *In this case, configure the server name including the domain name.*
>
> *e.g.: `terminal_01.mydomain.net`*

9.3 Special setups in the communication between Primary Server and Standby Server

Note the rules in the following setup:

▶ The Primary Server has failed

▶ The Standby Server has taken on its role

▶ The original Primary Server is restarted

▶ The original Primary Server gets all Runtime files from the current Primary Server (originally the Standby Server)

In exceptional cases, there may be conflicts if:

1. Changes have only been made to the project of the original Primary Server that has been stopped

2. It is not clear, due to network problems, which computer is the Primary Server.

1. PROJECT CHANGES WITH THE PRIMARY SERVER STOPPED

If, during the time in which Runtime is stopped on the original Primary Server, you make changes to the project and introduce these before synchronizing on this (stopped) computer, these changes are overwritten again as soon as the original Primary Server gets the data from the current Primary Server (the original Standby Server).

To prevent this: Introduce the amended data onto the Primary Server (configured Standby Server) as well, before starting the original (configured) Primary Server.

2. PROJECT CHANGES IN REDUNDANCY MODE OR THE SERVERS USED

The following changes are only accepted after the Primary Server has restarted:

▶ Change of the **Mode de redondance**.

▶ Changes in the **Serveur 1** properties or **Serveur 2** in the **Réseau** properties group.

> ⚠ **Attention**
>
> In this case, reloading the project in Runtime is not sufficient to apply the necessary changes! To accept all changes, restart the Primary Server.

3. SERVER ROLE NOT CLEAR DUE TO NETWORK PROBLEMS

In exceptional cases, it may happen that both computers are the Primary Server. The cause of this can be, for example, a loss of network connection due to a switch failure, a disconnected network cable etc. In this case, the communication between the Primary Server, Standby Server and Clients depends on the error screen.

If, with this setup, the error screen is resolved and both servers communicate with each other again (at this time), then the original (configured) server has data sovereignty. That means: The standby server's more recent data could be overwritten.

To prevent this:

1. Always check the role distribution with the **[Network] Current Primary Server** system variable: You will see the role that Runtime has and discover duplicate Primary Servers.

2. End zenon Runtime on the Primary Server that lost the network connection.

3. Set up the network connection again.

4. Restart zenon Runtime on this computer.

5. Runtime then starts the project with the computer as a Standby Server, updates its data and only then switches back to the Primary Server role.

Hint: Monitor the network connection with the Redundancy Management Tool (à la page 99).

> ⚠️ **Attention**
>
> *Les fichiers graphiques **PNG** ne peuvent pas être remplacés s'ils sont actuellement affichés dans le Runtime.*
>
> ***Arrière-plan :** Le Runtime protège les fichiers **.png** ouverts. Ceci évite qu'ils ne soient remplacés.*
>
> ***Solution :** Avant d'activer Remote Transport, assurez-vous que les synoptiques comportant des fichiers ***.png** :*
>
> ▸ Ne sont pas appelés dans le Runtime
>
> ▸ Ne sont pas utilisés par un autre programme
>
> *Cela s'applique également à l'actualisation des fichiers Runtime modifiés. La synchronisation dans le Runtime sur le réseau ne fonctionne pas pour un synoptique comportant un fichier ***.png** si l'appel de synoptique est effectué sur un ordinateur zenon impliqué dans le procédé (serveur redondant, client).*

9.4   Integrated evaluation methods for redundancy switching

In the dominant network, the Standby (as the Primary Server) regularly sends a telegram to the (dominant) Serveur 1, so that this - as soon as it receives a connection - can take on its role again. In a rated network, no telegram is sent any more in this case.

If there are two servers, the evaluation decides on the role.

USE OF DOMINANT BEHAVIOR IN ZENON LOGIC

▸ A driver can delay upgrading to a server on the dominant server. This function is no longer present in a rated network with severs that have equal rights.

▸ Archives also receive an extra byte per entry on the Standby Server as well as on the Serveur 2. This ensures that the archives are different and are included in a synchronization.

▸ Furthermore, the network topology for the drivers uses the terms Dominant and Standby as well as the roles as Server or Standby. However only the role is evaluated, so no change is therefore necessary.

▸ zenon Logic is supplied with information on the label and role via **CStratonVM::UpdatePrjSTates**. The only thing that is evaluated is whether a computer is configured as a Standby and not the Primary Server. in this case, zenon Logic is also not active.

9.5 zenon circular redundancy
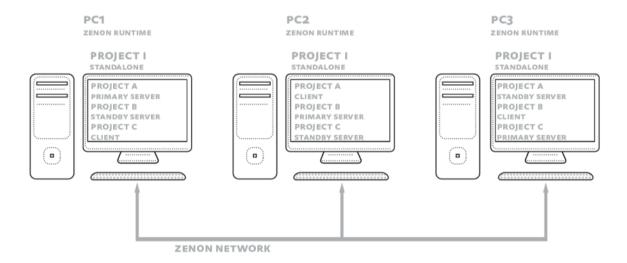
zenon circular redundancy allows seamless redundancy for several projects with a low amount of hardware being used.

Two computers are normally required for each redundant project: one computer as the Primary Server and one computer as the Standby Server. 3 projects would thus require 6 computers. Just three computers are sufficient for implementing three projects with zenon circular redundancy. Another computer is added for each further project. For this, zenon combines multi-project administration (à la page 30) with seamless redundancy (à la page 77).

CONCEPT OF CIRCULAR REDUNDANCY

Circular redundancy uses the possibility of multi-project administration. Several projects can run simultaneously on one computer. Each computer is the Primary Server for a project and the Standby Server for a second "neighboring project" and can also be the Client for further projects. This results in a circle. Instead of four computers, for example, and licenses for two projects, six for three or eight for four, you only need half of that.

Topology with three projects

▶ Computer 1 is the Primary Server for Project A and Standby Server for Project B.

▶ Computer 2 is the Primary Server for Project B and Standby Server for Project C.

▶ Computer 3 is the Primary Server for Project C and Standby Server for Project A.

   The circle is closed!

▶ Each computer can be a Client for the other projects at the same time.

▶ Expense: 3 computers and 3 Runtime licenses

> 💡 **Info**
>
> *An integration project (à la page 35) is needed to start more than one project on a computer.*

Normally you would need six computers and six Runtime licenses in this example. zenon circular redundancy is of course not limited to three projects, but can connect as many projects as desired in a circle. The fact that the computers can also be clients for other projects allows easy implementation of a low-cost, fail-safe, highly-available production line.

TIME SYNCHRONIZATION FOR ZENON CIRCULAR REDUNDANCY

If zenon time synchronization (à la page 18) is active, the Standby Server and clients always receive the current time from the Primary Server. This makes no sense when using zenon circular redundancy, because the individual PCs are Primary Server and Standby Server at the same time. Computer 1 for example, would thus obtain the time from computer 2, computer 2 would obtain it from computer 3 etc.

Recommendation: In this case, deactivate the zenon time synchronization and carry out external time synchronization. You can find instructions for this in Time synchronization in the network (à la page 18).

9.6 Redundancy Management Tool

The **Redundancy Management Tool** monitors the network adapter and its connection to the network. If the device loses the connection to the network - e.g. by removing the network cable, the **Redundancy Management Tool** stops the Runtime. This process can be canceled by the operator within a configurable period of time. If the connection to the network is reestablished, the Redundancy Management Tool restarts the Runtime.

START AND CONFIGURATION

The Redundancy Management Tool can be configured as an application with a dialog or by means of the command line interface.

To open the dialog:
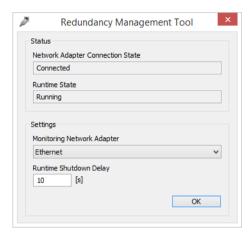
▶ Using the Windows start folder:
Start -> All programs -> COPA-DATA -> Tools 7.50 -> **Redundancy Management Tool**

▶ Using the Startup Tool:
Tools -> zenon_redman

▶   Direct start of the file zenon_redman.exe from the zenon program folder

After the start the **Redundancy Management Tool** is also displayed as symbol in the right area of the Windows task bar. A Double click on the symbol opens the configuration dialog:

STATUS

Status information on the monitored network adapter.

Parameters	Description
Network Adapter Connection State	Information about the status: ▶ `Connected:` Connection to the network established. ▶ `Disconnected:` Connection to the network interrupted.
Runtime State	Status of the zenon Runtime ▶ `Running:` Runtime is running. ▶ `Stopped by Redundancy Management Tool:` Runtime was closed by the tool. ▶ `Stopped:` Runtime is not running.

SETTINGS

Configuration of the network adapter to be monitored

Parameters	Description
Monitoring Network Adapter	Selection of the network adapter which should be monitored from the drop-down list. The list shows all network adapters available on the computer.
Runtime Shutdown Delay	Setting of the delay time before ending Runtime, in seconds. In this period of time the operator can cancel the closing of the Runtime. Default: `10 s` Maximum value: `2147483647 s` Values above this are interpreted as `0`.
OK	Applique les paramètres et ferme la boîte de dialogue. Settings are applied in the INI file.

INI FILE

At the configuration via the dialog, file RedMan.ini is created in path %ProgramData%\COPA-DATA\System. It contains the following entries.

Parameters	Description
[DEFAULT]	
ADAPTER=	Selected LAN connection.
DELAY=	Value for delay time.

COMMAND LINE

The Redundancy Management Tool can also be started via the command line.

Possible parameters:

▸ **ADAPTER=**[Name]
Defines the network adapter which should be monitored.

▸ **DELAY=**[Seconds]
Defines the waiting time after a connection loss.
Maximum value: 2147483647. Values above this are interpreted as 0.

▸ HELP,?
Displays help about the command line parameters

> **Informations**
>
> *At the configuration via command line:*
>
> ▸ these settings are taken over directly
>
> ▸ the configuration is deactivated in the dialog
>
> no INI file is written
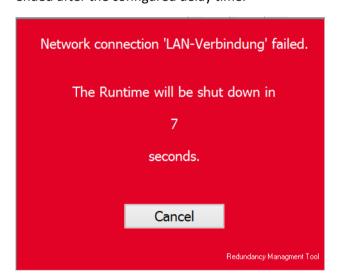
IN THE RUNTIME

During the Runtime the **Redundancy Management Tool** monitors continuously the network connection. If the connection is canceled, the **Redundancy Management Tool** shows a warning message. Runtime is ended after the configured delay time.

Network connection 'LAN-Verbindung' failed.

The Runtime will be shut down in

7

seconds.

Cancel

Redundancy Managment Tool

As soon as the connection is available again, the **Redundancy Management Tool** restarts the Runtime.

Click on button **Cancel** to halt the countdown and prevent the closing of the Runtime. If the connection is reestablished, the dialog is displayed again when the connection fails again. The user can cancel again or let the tool close the Runtime.

> **Informations**
>
> The current status of the connection and Runtime is also always displayed in the configuration dialog.

ERROR HANDLING

ERROR MESSAGE

Error are displayed by pop-up messages.

Message	Meaning
`GetAdapterAddresses not supported on this platform! Error code '%u'!`	Operating system version is not supported
`GetAdapterAddresses did not return information about network adapters. Error code '%u'!`	No network adapter found.

LOG FILES OF THE DIAGNOSIS VIEWER
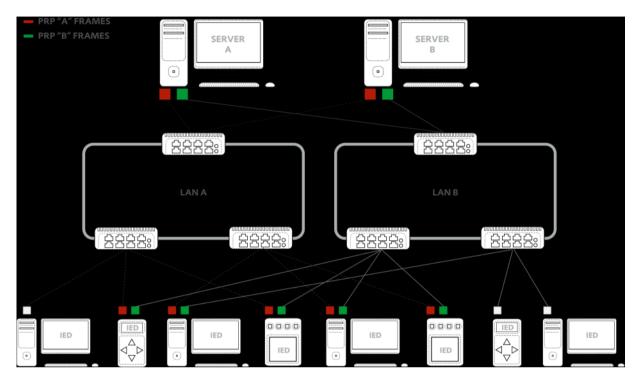
In the log file of the Diagnosis Viewer the following is documented:

Entry	Debug Level	Meaning
`Network link '%s' down for '%u' seconds. zenon Runtime will be terminated.`	Error	Network connection failed: The Runtime is closed.
`Network link '%s' is up. Restarting zenon Runtime now.`	Information	Network connection available again: The Runtime is restarted

10. COPA-DATA PRP

zenon supports the Parallel Redundancy Protocol (PRP) for hardware-redundant communication in a network.

PRP communication is carried out at at protocol level directly, regardless of the zenon Editor and zenon Runtime. Special configurations in zenon are not required. To use the protocol, the computer must be configured accordingly.

You can find the network service COPA-DATA PRP, necessary for the use of PRP, on the installation medium. You can find a detailed description of the required configuration steps in this chapter in the installation and configuration (à la page 105) chapter.

 Informations concernant la licence

The network service is supplied.
A valid zenon license is required on the computer for use.

10.1   System requirements

PRP communication is supported for the following operating systems:

▶   Windows 7

▶   Windows 8

> ⚠ **Attention**
>
> *Windows 10 is not supported for PRP communication due to the system.*

10.2   Hardware requirements

The following hardware requirements are applicable for communication via PRP:

▶   The network card use must support jumbo frames

▶   Configuration of the locally-administrated MAC address must be possible on the network card used.

> ⚠ **Attention**
>
> PRP communication is only supported in a redundant network.

10.3   Installation and configuration

To use PRP communication, carry out the following configuration steps.

1.   Disconnect the computer with the two network cards from the power supply before installation.

2.   Restart the computer

3.   Configure your network adapter.

4.   Create a network bridge in the system settings (**Bridge**).

5.   Install the `COPA-DATA PRP driver`.

6.   Configure your PRP connection

PLEASE KEEP IN MIND:

▶   Administrator rights on the computer are required for installation.

▶   The system must be restarted for the installation.

▶   Note the instructions for the respective steps.

> ⚠ **Attention**
>
> *Ensure that you carry out the configuration steps in the given sequence.*

10.3.1 Configuration of network adapter (step 1 of 4)

In the first step, amend the configuration for both network adapters used. The configuration dialog and the naming of the enhanced properties depends on the network card.

NETWORK ADAPTER 1

Configure the first network adapter.
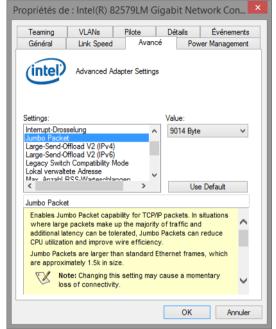
1. Open the **Change adapter settings** system setting.
   You can find these settings in the `Control Panel => Network and Internet => Network and Sharing Center`

2. Select the desired network adapter.

3. With the right mouse button, select the **Properties** entry in the context menu.
   The configuration dialog for the properties of the network adapter are opened.

4. Click on the **Configure ...** button
   The properties window of the network adapter is opened.

5. Switch to the **Advanced** tab there.

6. In the list of **settings** there, select the **Jumbo Packet** entry
   Note: The name of this entry may be different for each network card.

7. Select a value in the **Value** drop-down list.
   Select the lowest-available value that is greater than 1530 bytes.
   Attention: The Disabled setting must not be selected.

8. In the **Advanced** tab, select the **Locally-administered address** setting.

9. Enter a unique MAC address in the `Value:` input field. The format of the MAC address depends on the hardware used.
   Examples:

   - 0A:80:41:ae:fd:7e

   - 0A-80-41-ae-fd-7e

   - 0A8041aefd7e

10. Ensure that, for both connections used, the same MAC address is used.
    Change this address in the **Value** input field:

    - This MAC address must start with `0A`!

    - The MAC address in the local network must be unique.

11. Finish configuration of the network card by clicking on the **OK** button.

NETWORK ADAPTER 2

Repeat the steps for the second network adapter.
When entering the MAC address, ensure that the same MAC address as the one in the previous configuration is entered.

⚠ **Attention**

Ensure that
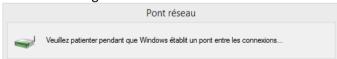
▸ The MAC address used on both computers is the same

▸ It is not used by any other computer in the local network.

10.3.2 Bridge network adapter (step 2 of 4)

In this step, you combine two network adapters with a network bridge. Amend the configuration for both network adapters used.

Create a network bridge in the system settings.

1. Open the **Change adapter settings** system setting.
   You can find these settings in the `Control Panel => Network and Internet => Network and Sharing Center`

2. Select the two network adapters that you want to use for PRP communication.
   Note: Note that for both network adapters, the necessary configurations have already been carried out. A subsequent amendment to the configuration of a network adapter only becomes effective if you then create a new bridge.
   Attention: Both network adapters selected must be configured with the same MAC address!

3. With the right mouse button, select the **Bridge connections** entry in the context menu.
   A network bridge is created for the selected network adapter. This is visualized in a dialog.

4. The bridge created is displayed in the Control Panel:

   Attention: The bridge must only contain two adapters.
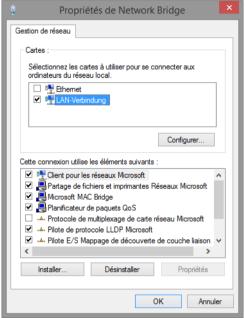
10.3.3   Installation of COPA-DATA PRP driver (step 3 of 4)
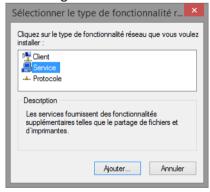
In this step, you install the service system required for PRP communication.

Install the `COPA-DATA PRP driver`

1. Select the bridge created.

2. With the right mouse button, select the **Properties** entry in the context menu.
   The configuration dialog for the properties of the bridge is opened.
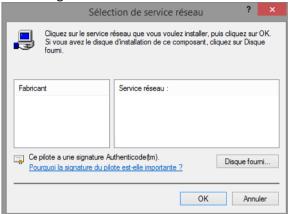
3. Click on the **Install** button.
   The dialog to install a network feature is opened.
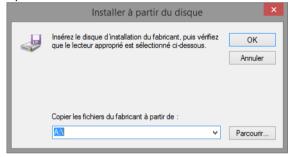
4. Select **Service** as the network feature to be installed.

5. Click on the **Add...** button
   The dialog for the selection of the network service is opened.

   Sélection de service réseau
   Cliquez sur le service réseau que vous voulez installer, puis cliquez sur OK.
   Si vous avez le disque d'installation de ce composant, cliquez sur Disque fourni.

   Fabricant

   Service réseau :

   Ce pilote a une signature Authenticode(tm).
   Pourquoi la signature du pilote est-elle importante ?

   Disque fourni...

   OK   Annuler

6. Click on the **Data medium ...** button
   The dialog to select the save location of the installation program for the network service is opened.
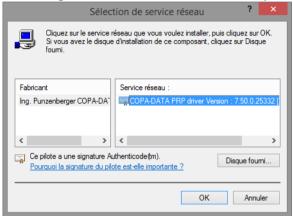
   Installer à partir du disque
   Insérez le disque d'installation du fabricant, puis vérifiez que le lecteur approprié est sélectionné ci-dessous.

   OK
   Annuler

   Copier les fichiers du fabricant à partir de :
   A:\

   Parcourir...

7. Click on the **Browse** button.

8. Go to the following folder on your local computer:

   - `\Programs (x86)\Common Files\COPA-DATA\CDPrpFlt\`
     for 32-bit operating systems.

   - `\Programs\Common Files\COPA-DATA\CDPrpFlt\`
     for 64-bit operating systems.
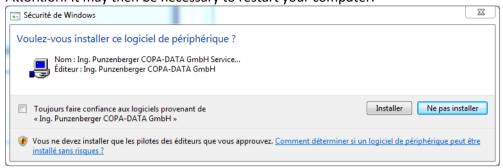
9. Select the `CDPrpFlt.inf` file.
    Attention: Ensure that you select the correct installer for your operating system (32-bit or 64-bit).

10. Confirm the selection by clicking on **OK**.
    The dialog to select the network service is opened.

11. Select the `COPA-DATA PRP driver` network service.

12. Confirm your selection with **OK**.
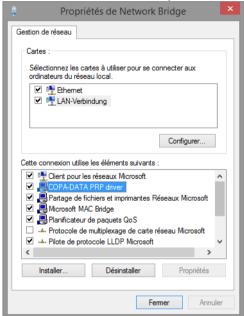
- Confirm the Windows request for confirmation by clicking on the Install button.
  Attention: It may then be necessary to restart your computer.

Note: This request for confirmation is not shown if you have already activated the " . . . `always trust`" box when installing zenon program components earlier.

13. After successful installation (and restarting the computer) the service is visible in the properties window of the network adapter in the list of elements used.

14. Ensure that the LAN connection and the network service **COPA-DATA PRP driver** are activated using the checkbox.

> ⚠ **Attention**
>
> *Ensure that use in the active system is not jeopardized by the required restart.*

10.3.4   Configuration of PRP connection (step 4 of 4)
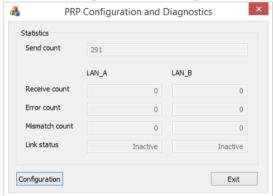
Before configuration, ensure that the LAN connection and the **COPA-DATA PRP driver** network service are activated.

PRP CONFIGURATION

1. Start `PRPCfgDiag.exe`.
    You can find this software on your computer in the `C:\Program Files (x86)\Common Files\COPA-DATA\STARTUP` folder.
    The **PRP Configuration and Diagnostics** dialog is opened.
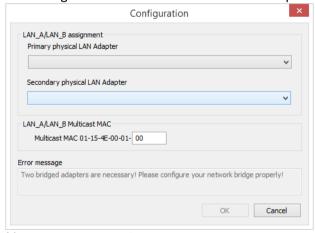
    Note: The **PRP Configuration and Diagnostics Tool** is only available in English.

2. Click on the **Configuration** button.
    The dialog for the selection of the network adapter is opened.

    Note: The content of the drop-down list is based on the system settings.

3. Select, from the drop-down list, the network adapter for **LAN_A** and **LAN_B**.

4. Confirm the assignment with **OK**.

5. End the configuration by clicking on the **Exit** button.
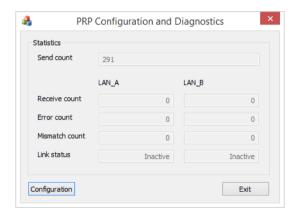
10.4   PRP configuration and diagnosis tool

The **PRP Configuration and Diagnostics Tool** performs two tasks:

▶ Visualization (à la page 115)
   Display of the data traffic sent via PRP. The display is separate for the two network adapters used.

▶ Configuration (à la page 116)
   Assignment of the configured network adapter.

Remarque : This dialog is only available in English.

PRPCfgDiag.exe is supplied with zenon.
You can find this software on your computer in the `C:\Program Files (x86)\Common Files\COPA-DATA\STARTUP` folder.

REQUIREMENTS

The **PRP Konfigurations- und Diagnose Tool** needs the following for operation or configuration:

▶ Two network adapters that are combined into a bridge in the system settings.
   Note: In this bridge, only the two network adapters that are used for PRP communication can be configured. Other network adapters must not be included in this bridge.

▶ The CDPrpFlt driver must be installed.
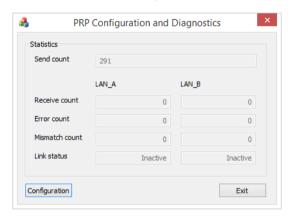
<table><tr><td>🔆</td><td>**Informations**</td></tr></table>

You can find information on the installation and necessary preparations in the system settings in the installation and configuration (à la page 105) chapter.

10.4.1   Statistics

The data flow is visualized in the **Statistics** dialog. The setting is displayed separately for both LAN adapters.

The flow of data is always recorded, even if the tool is not open.

Remarque : This dialog is only available in English.

Parameters	Description
Send count	Display of the Ethernet frame sent.
Receive count	Display of the Ethernet frame received.
Error count	Display of invalid PRP frames.
Mismatch count	Display of PRP frames received/sent differently if the network data traffic of the two LAN adapters differs from one another.
Link status	Status of the network card: ▶ `Active` Assigned network card is active and data flow is transferred. ▶ `Inactive` Assigned network card is not active. No PRP Supervision frames are received. There is no PRP station in the network or there is an error.
Configuration	Opens the configuration dialog (à la page 116).
Exit	Closes the program. Note: The data continues to be recorded.

10.4.2   Configuration

The following takes place in the configuration dialog:

▶ Network adapter is assigned by means of a drop-down list.
The content of the drop-down list is based on the network settings.
You can find further information in the installation and configuration (à la page 105) chapter.
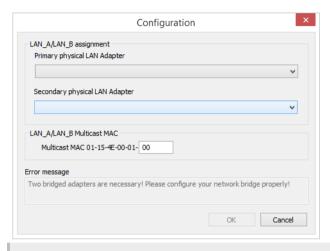
▶ The multicast MAC address is visualized

▶ Error messages from the network adapter configuration are visualized in an output window

⚠ **Attention**

The computer must be restarted after changes to the configuration have been made.

Remarque : This dialog is only available in English.

Parameters	Description
Primary physical LAN Adapter	Assignment of a network adapter to the physical connection for the primary LAN adapter.
	In the drop-down list, the adapters that are included on the configured bridge are listed.
	You can find information on this in the installation and configuration (à la page 105) chapter.
Secondary physical LAN Adapter	Assignment of a network adapter to the physical connection for the secondary/redundant LAN adapter.
	In the drop-down list, the adapters that are included on the configured bridge are listed.
	You can find information on this in the installation and configuration (à la page 105) chapter.
LAN_A/LAN_B Multicast MAC	Multicast MAC address for PRP-Supervision frames. This address for communication in the network is preset and cannot be changed.
	Note: Ensure that no other network adapter in your network uses this address!
	The last byte can be configured in the input field. The input format for this entry is HEX.
Error message	Output window with error messages.
OK	Accepts all changes and switches to statistics dialog (à la page 115).
Cancel	Discards all changes and switches to statistics dialog (à la page 115).

11. Routing

It is strongly recommended that the **Routage actif** project is no longer used for a current project configuration. This property is deactivated by default if the zenon network is reconfigured.

PROCEDURE

For routing, the packets of subordinate projects are sent to the first client project (**FCP**) in the branch. The computer acts as node computer and can route packets.   Thereby all network packets from the outside use this computer. However, this setting can lead to bottlenecks and influences the possible

network topology. It is only worthwhile using it in special network setups, e.g. for slow WAN networks or routed networks.

▶ Example:
If, in a setup consisting of several computers, not all computers can reach the others, a computer can act as a router.

▶ Technical implementation:
The **Serveur 1** and the **Serveur 2** of the subordinate projects are amended on that of the FCP; this is the **Serveur 1/Serveur 2** active in runtime.

11.1   General notes on routing

Two basic rules must be noted when configuring network structures with routing. If one of these rules is not adhered to, communication problems or other undesired effects may occur depending on the respective structure.

▶ Rule 1: Server and levels
A PC that acts as a server may only in one level (circular redundancy) act as a server or Standby several times. It must not be defined as a server a level above or below.

▶ Rule 2: Standalone

If the start project is a single-user project, only one single level below can be used for network projects.

CLIENT SENDS TO A SERVER

▶ The client sends the packet to the server active in the project in Runtime.

▶ If the project on this computer is not the server, the packet is sent until it arrives at the server.

▶ This functionality is not affected by an integration project.

SERVER SENDS TO A CLIENT WITH ROUTING

1. If the server has a direct connection to the client, the packet is sent there.

2. If there is no connection to the target computer, the server sends the packet to all computers on which the project is running for which it acts as a server.

3. If the node has a direct connection to the client, the packet is sent there.

4. If the computer works as a node, then the packet is sent to all computers which have connected to the node computer. It the target computer is also the source computer, the packet is not sent any further.

5. The procedure is continued at point 3.

Note: Points 2 and 4 are only carried out if routing is active on these computers.

> 💡 **Informations**
>
> *The Server and Standby need not correspond to what has been configured on the client computers, otherwise they may change themselves depending on the topology of the respective computer.*

WHAT IS A CLIENT CONNECTION?

A network service connection is labeled as a client connection if it is made to the server or standby handling the process by a client. This is recognizable in that there is a connection to port 1100 on the target computer.

> ⚠ **Attention**
>
> *It is not guaranteed that that a pure client computer added to a functional, defined topology will work. It is possible that some projects cannot be reached by the server due to routing on client computers in particular.*

CHECKING THE ROUTING

To check the routing settings, use the procedure from "Administering network topology (à la page 67)".

11.2 Compatibility

RULES FOR ROUTING BEFORE ZENON VERSION 6.50:

1. The first client network project of a branch on a computer defines the server and standby for all subordinate projects in the branch. This also applies

   - If a subordinate project on this computer is a server or a standby server

   - for projects that do not really have a Standby Server

2. If the subordinate project is not a network project or is not a server, the branches of the subprojects of the start project are considered in parallel. Different computers can therefore be servers for the subprojects. The rules from item 1 apply for the branches.

3. Single user projects are not taken into account for the topology, with the exception of the start project.

4. If the start project is not a server (i.e. single user, client or standby not handling the process), routing is not activated in the network service. This only affects the direction from the server to the client.

RULES FOR ROUTING FROM ZENON VERSION 6.50:

The **Routage actif** property is deactivated as standard from version 6.50 onwards.

WITHOUT ROUTING

If the **Routage actif** property is not active for the start project on the computer, routing does not take place. Each project then connects directly to the corresponding computer, where it is the server. The computer is then not a node and packets are also not routed from here.

WITH ROUTING

The rules as they were prior to 6.50 remain valid.

Exception:

  ▶ A project that is a server or standby on the computer remains a server or standby, even if the superordinate project uses another server or standby.

12. Authorization in the network

A network project can be operated from all stations in the same way with the basic settings.

Operating here means actively intervening in the process, such as:

  ▶ Send values

  ▶ Write recipes

  ▶ Acknowledge alarms

  ▶ etc.

There is thus the danger that two users on two different stations want to set different values for the same variable at the same time.

In this case:

  ▶ Both actions are executed

  ▶ The values that is entered last overwrites all previous ones

In zenon you have the possibility to allow operation of the project only from one station at the time. In this case the operator has to get the authorization, before he can operate the project.   Opening screens, as well as read access to lists such as AML, CEL, recipes, etc. is possible on each station as always.

> 💡 **Informations**
>
> *Operating authorizations for projects without a network can be implemented by evaluating a binary variable for the project property* ***Vérouillage fonctionnement****. For details, see the Operating authorizations chapter in the Project administration and workspace manual.*

PROCEDURE

The following is applicable if the **Autorisation en réseau active** property is active:

- ▶ Authorization must be obtained if active operation takes place.
- ▶ If operation is blocked by another computer, a dialog is opened on the compute that is blocking it.
- ▶ The user who is locking it can approve the authorization or keep it locked.
- ▶ If there is no response, the authorization is approved after a pre-set timeout.
- ▶ If an interruption in the network connection is recognized, then the authorization for this computer is reset.

For details see chapter:

- ▶ Configuring authorization (à la page 123)
- ▶ Authorization in Runtime (à la page 124)

SYSTEM VARIABLES FOR AUTHORIZATION

The system variables inform you about authorization:

- ▶ `[Network] Computer with authorization`:
  Name of the computer that has the authorization (Data type: `String`)
- ▶ `[Network] Authorization present for this computer`:
  Computer has authorization (Data type: `Bool`)
- ▶ `[Network] Authorization denied`:
  Computer requests authorization, but does not receive it (Data type: `Bool`)

  > 💡 **Informations**
  >
  > *You can find further information in the SYSDRV manual in the Theme [Network] chapter.*

12.1 Configuring authorization

To enable authorization in the network, you must:

- ▶ Activating authorizations and setting timeouts
- ▶ Configuring functions for getting and approving operating authorizations in Runtime

ACTIVATING AUTHORIZATIONS

The operating authorization in the network must be activated in the project properties.

- ▶ navigate to the **Réseau** group in Project Properties.

- ▶ Activate the **Autorisation en réseau active** property.

- ▶ Define the **Timeout pour requête [s]**:

  Defines the period of time in which a computer can respond to an approval request. The authorization is automatically approved after this time has expired.

  Default: 60 seconds

- ▶ Define the **Timeout pour autorisation [s]**:

  Defines the period of time in which a computer that has authorization must report to the Primary Server. The authorization is automatically approved after this time has expired. Connection interruptions in the network are therefore recognized. The authorization can therefore not be blocked by a compute that cannot be contacted.

  Default: 60 seconds.

  Attention: Select the time period as shorter than the network timeout.

FUNCTIONS FOR AUTHORIZATION IN RUNTIME

To obtain authorizations or to approve these, the corresponding functions must be available in Runtime. To do this, create two buttons that are designated for the corresponding functions:

- ▶ `Get authorization`:
  Obtains authorization from the user's own computer

- ▶ `Approve authorization`:
  Approve authorization or explicit request

GET AUTHORIZATION

1. Create a new function.

2. Select the **Network** function in the **Autorisation sur le réseau** group.

3. The selection dialog for authorizations in the network is opened.

4. Select **Get**.

If this function is executed in Runtime, the authorization can be obtained from the user's own station.

APPROVE AUTHORIZATION

1. Create a new function.

2. Select the **Network** function in the **Autorisation sur le réseau** group.

3. The selection dialog for authorizations in the network is opened.

4. Select **Approve**.

If this function is executed in Runtime, the authorization can be approved again.

12.2 Authorization in Runtime

If the **Autorisation en réseau active** property is active, active operations are only executed in Runtime if there is operating authorization for the station.

EXAMPLE

If there is no operating authorization, a set value should be written to a variable:

1. The set value is not sent to the hardware when the button is clicked on.

2. Instead, a message box opens informing you that you do not have the authorization for this project.

3. Click on the **Obtain the operating authorization** button that has been configured.

DOES NOT BLOCK AUTHORIZATION FOR ANY OTHER COMPUTER:

If the operation is freely available:

- ▶ You receive the authorization

- ▶ You can write the set value

- ▶ You can, after the operation, make this available to others using the **Approve authorization** function

BLOCKS AUTHORIZATION FOR ANOTHER COMPUTER:

If authorization is blocked:

- ▶ A dialog is opened on the computer that is blocking

- ▶ The user of the computer that is blocking must explicitly release the authorization

  - • **Yes**: The authorization is passed over to the other computer

  - • **No**: Authorization remains blocked

  - • **No reaction**: Countdown corresponding to the timeout time defined in the **Timeout pour autorisation [s]** property runs out. The authorization is automatically released at 00:00.

13. zenon functions in the network

Special zenon functions for the network:

- ▶ **Authorization in network** (à la page 126)

- ▶ Redundancy switch

  - a) in redundancy mode: **Dominant** (à la page 126)

  - b) in redundancy mode: **Non-dominant**

  - c) in redundancy mode: **Rated**

In general, the location of execution (à la page 136) must be noted when using functions in the network. For some functions, the location of execution can be freely configured; this is fixed for others.

13.1 Authorization in network

To obtain authorizations or to approve these, the corresponding functions must be available in Runtime. To do this, create two buttons that are designated for the corresponding functions:

- ▶ `Get authorization:`
  Obtains authorization from the user's own computer

- ▶ `Approve authorization:`
  Approve authorization or explicit request

GET AUTHORIZATION

1. Create a new function.

2. Select the **Network** function in the **Autorisation sur le réseau** group.

3. The selection dialog for authorizations in the network is opened.

4. Select **Get**.

If this function is executed in Runtime, the authorization can be obtained from the user's own station.

APPROVE AUTHORIZATION

1. Create a new function.

2. Select the **Network** function in the **Autorisation sur le réseau** group.

3. The selection dialog for authorizations in the network is opened.

4. Select **Approve**.

If this function is executed in Runtime, the authorization can be approved again.

13.2 Redundancy switching in the dominant network

With this function, manual switching between the Primary Server and Standby Server is possible. The current Primary Server thus becomes the Standby Server and vice versa.

The function is available for all redundancy modes (zenon Editor **Mode de redondance** property):

- ▶ Non-dominant
- ▶ Dominant
- ▶ Rated

Note: The redundancy switching function is only available if the **Réseau actif** property has been activated.

APPLICATION EXAMPLES

Possibilities for use in practice are:

- ▶ Planned maintenance work on the Primary Server
- ▶ Better hardware coupling of the Standby Server
- ▶ In an evaluated network:
  Deactivate evaluation temporarily

> ⚠ **Attention**
>
> *This function is not suitable for testing redundancy, as the behavior differs from that of a server failure.*

SWITCHING DELAY

Some zenon modules can delay redundancy switching. For example, a **command** that is running delays redundancy switching.

Please note the module descriptions in the Behavior of zenon modules in the network (à la page 130) in this manual in relation to this.

13.2.1 Project configuration in zenon

CREATE REDUNDANCY SWITCHING FUNCTION

A **Commutation redondance** function serves to control the roles of **Primary Server** and **Standby Server** in a zenon Network.

ENGINEERING

Steps to create the function:

1. Create a new function:
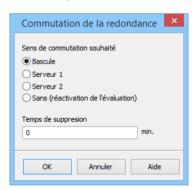
   In the tool bar or in the context menu of the Functions node, select the **New function** command. The dialog to select a function is opened.

2. Navigate to node **Réseau**

3. Select the **Commutation redondance**.
   The dialog to configure redundancy switching (à la page 128) is opened.

4. Configure the behavior of redundancy switching.

5. Name the function in the **Nom** property.

👍 **Conseil**

Configure a separate redundancy switching function for each switching direction.

13.2.2   Configuration dialog for redundancy switching

You configure the action of a planned **Commutation redondance** in the **redundancy switching** dialog.

```
Commutation de la redondance                    ×

Sens de commutation souhaité
◉ Bascule
○ Serveur 1
○ Serveur 2
○ Sans (réactivation de l'évaluation)

Temps de suppresion
[0                    ]  min.


[   OK   ]  [  Annuler  ]  [   Aide   ]
```

SWITCHING DIRECTION

Parameters	Description
Toggle	**Serveur principal** and Standby Server switch roles.
Serveur 1	**Serveur 1** becomes (or remains) **Serveur principal**.
Serveur 2	**Serveur 2** becomes (or remains) **Serveur principal**.
Without (reactivation of the assessment)	No switching takes place. Instead, a possible switching for the **suppression time** configured in the properties field is prevented for the configured duration. If the **suppression time** is 0, the switching is possible again immediately. Note: This option is only available in a rated network.

SUPPRESSION TIME

Parameters	Description
Suppression time	The time in minutes, within which no automatic redundancy switching takes place due to an amended evaluation. If the value is 0, the automatic switching is carried out as configured in the evaluation principles. Range of values: 0 to 10080 minutes (= one week) This means: The current Primary Server (**Serveur 1** or Server 2) remains the Primary Server for this time, regardless of the current result of the evaluation. If **Temps mort après commutation [s]** and the **suppression time** are configured differently in a rated network, the greater value prevails here. Note: This option is only available in a rated network.

NAVIGATION

Parameters	Description
OK	Applique les paramètres et ferme la boîte de dialogue.
Cancel	Annule toutes les modifications et ferme la boîte de dialogue.
Help	Ouvre l'aide en ligne.

💡 **Informations**

Parameters	Description
You can find further information in the Switching delay, down time and hysteresis (à la page 86) chapter in this manual.	

14. Behavior of zenon modules in the network

With network projects, the behavior of individual modules and functions in the network should be noted.

14.1 AML and CEL

ALARMING

The alarming is administered on the Primary Server. The Primary Server answers requests for alarming from the clients. Changes are synchronized between the Primary Server and Standby Server.

Informations

Si le serveur de secours reprend le rôle de serveur principal suite à une défaillance du précédent serveur principal, les données manquantes dans les listes AML et CEL et les archives sont renseignées. Les données manquantes proviennent du buffer interne du serveur de secours. Ce buffer est alimenté en valeurs par les drivers.

CHRONOLOGICAL EVENT LIST

The CEL is administered on the respective Primary Server. Changes are synchronized between the Primary Server and the Standby Standby Server.

In a rated network, no new CEL entries are created for regrading due to the evaluation.

> ### 👍 Conseil
>
> *In order to log the evaluations in the CEL, create a system driver variable "**[Network] Evaluation Result of Server 1**" or "**[Network] Evaluation result of Server 2**":*
>
> ▸ Select, in the **Espace de travail**, in the **Variables** node, the **New variable...** entry
>
> ▸ Select `SYSDRV – driver for system variables` as a **driver**.
>
> ▸ In the **Variables of the system driver** dialog, select the `Network` entry in the **Theme** entry.
>
> ▸ Select **[Network] Evaluation Result of Server 1** and **[Network] Evaluation Result of Server 2** and add these to the list of system driver variable to be created by clicking on the **Add** button.
>
> ▸ Create a numerical reaction matrix, activate the "**Treat each change of value as new limit violation**" and set the "**In Chronological Event List**" option
>
> ▸ Link this newly-created reaction matrix to the two variables.

14.2   Serveur d'archives

L'archivage est effectué sur le serveur principal.

Le serveur principal synchronise les données de l'archive avec le Serveur redondant et répond aux demandes des clients (par exemple, appel d'un synoptique `Extended Trend`).

> ### 💡 Informations
>
> Si le serveur de secours reprend le rôle de serveur principal suite à une défaillance du précédent serveur principal, les données manquantes dans les listes AML et CEL et les archives sont renseignées. Les données manquantes proviennent du buffer interne du serveur de secours. Ce buffer est alimenté en valeurs par les drivers.

COMMUTATION DE LA REDONDANCE

Lors de la commutation de la redondance, l'activation de la commutation par le module Serveur d'archives de zenon peut entraîner un délai de commutation.

Si des variables issues d'un autre projet sont enregistrées dans une archive, le comportement au démarrage du Runtime sur le réseau de zenon change. Dans ce cas, les archives sont uniquement démarrées après le chargement de tous les projets.

Ceci permet au système de s'assurer que toutes les variables devant être archivées sont détectées avant le début de l'archivage et que l'ordinateur reprend le rôle de serveur principal.

> **⮞ Exemple**
>
> Le Runtime de zenon est démarré sur un ordinateur configuré en tant que **Serveur 1** ou **Serveur 2** durant la configuration du projet.
>
> ▶ Le Runtime démarre dans le rôle du serveur de secours.
>
> ▶ Tous les projets sont chargés.
>
> ▶ L'archivage est comparé.
>
> Les étapes sont exécutées, quel que soit l'évaluation ou le rôle actuel de l'ordinateur.
>
> Lorsque ces étapes ont été effectués, la commutation de redondance est effectuée (uniquement si nécessaire).

14.3 Batch Control

The module **Batch Control** is fully capable of using a network in terms of Client/Server technology. This means that Batch recipes can be created, duplicated, edited, deleted, etc. on a Client. The whole recipe management remains always on the server. Likewise the whole process control such as **start** recipe, **pause** recipe, **stop** recipe, etc. can be done from the Client. Also mode changes and manual operations such as **jump** are possible.

> **⚠ Attention**
>
> *Module Batch Control does not support redundancy. There is no synchronization between Standby Server. When the Server breaks down, the executed Batch recipes are not continued seamlessly on the Standby!*

For using Batch Control in a network the following is true:

ALLOCATION

▶ The forcing of allocations can be carried out from the Server or Client.

FUNCTIONS

Functions are always carried out at the Server.

PHASES

▶ Editing phases in the master recipe:

- Edit mode: Changes a done locally at the Client.
  If during the editing the recipe is saved on another computer in the network, the current configuration is lost. An appropriate message is displayed and the editing dialog is closed. The new data from the server are displayed.

- Test mode: Changes a done at the Server.

▶ Control recipe: Changes a done at the Server.

▶ If a recipe is saved in the network, all Clients using this recipe are updated.

▶ If a recipe is opened on a client, the current version on the server is always displayed, even if it has not yet been saved there.

▶ If a recipe is deleted on a computer, a message is displayed on all computer on which the recipe is opened that the recipe has been deleted.

MODE

▶ The mode (automatic, semi-automatic, manual) can be switched by the server and the client.

▶ Jumps in the recipe and step-by-step progress of a recipe can be done from Server and Client.

RECIPES

▶ Recipes can be started and controlled by the zenon server and by zenon clients.

▶ If parameters in a recipe are changed whilst the recipe is saved on a different zenon client, the change to the parameters is refused and not carried out.

▶ A master recipe can be changed on the zenon client whilst it switches to test mode on the zenon server and is sent to the zenon client. The changes that were last saved are transferred. This means: If the zenon client saves last, the recipe is switched to editing mode again. If the zenon server saves last, the change to the zenon clients is discarded and the recipe is in test mode.

▶ If a communication error occurs when deleting a recipe or an operation template, the deletion is refused with an error message.

WEB CLIENT

With a standard web client:

▶ The settings for grid and color can be changed

▶ No recipes can be created or edited

▶ The size of the editing area cannot be changed

▶ In the toolbar, all symbols that are not permitted are deactivated; it is not possible to select the corresponding objects.

Web client PRO is not affected by these restrictions.

14.4   User Administration

User administration is administered on the Primary Server. Log-in procedure:

1. The login request is sent to the Primary Server.

2. It answers with the list of authorized users.

3. The client verifies the data.

If changes to user administration are made on a client in Runtime, the complete user list is sent from the client to the Primary Server.

♀ Info

Active Directory, ADAM, ADLDS users

If Active Directory, ADAM or ADLDS users are used, all computers (regardless of whether they are the Primary Server, Standby Server or Client) communicate directly with the Active Directory, ADAM or ADLDS server.

This means that all computers must be in the corresponding infrastructure (such as Active Directory domains when Active Directory users are used); it is not sufficient that only the Primary Server is in the Active Directory domains with the corresponding users.

14.5   Files

Lists for the files of all modules are created when data is exchanged between the Primary Server and the Standby Server. The Primary Server monitors these lists for changes. Changes that are detected are transferred to the Standby Server.

⚠ Attention

The Primary Server does not react to watchdogs that are sent by the Standby Server when lists are created. Note this when stipulating the time for the network timeout.

Avec le module Remote Transport, tous les fichiers requis pour le projet sont transférés vers le système cible.

Tous les fichiers sont systématiquement transférés vers le dossier correspondant.

▶   Standard

Tous les fichiers présents dans le dossier de Runtime du projet (**...\RT\FILES\zenon\system**).
Ces fichiers déterminent l'apparence et le comportement du projet, et sont transférés par
défaut :

Informations concernant 💡

Les fichiers comportant les suffixes suivants ne sont pas transférés par défaut :

▸ .hot

▸ .ho

▸ .ret

▸ .re

▸ Facultatif

En outre, tous les fichiers incorporés dans le projet doivent être transférés. Pour les sélectionner,
cochez la case **Active** dans les paramètres de Remote Transport. Ces fichiers se trouvent dans les
sous-dossiers suivants du répertoire du projet :

- `\zenon\custom\graphics` : pour les éléments graphiques

- `\zenon\custom\lists` : pour les tables de langues

- `\zenon\custom\media` : pour tous les fichiers média

- `\zenon\custom\reports` : pour les rapports créés avec Report Generator

- `\zenon\custom\help` : pour les fichiers d'aide

- `\zenon\custom\additional` : pour les fichiers supplémentaires

- `\zenon\custom\additional` : pour les fichiers Report Viewer

- `\zenon\custom\drivers` : pour les drivers

- `\straton` : pour zenon Logic

Recommandation : Le chemin d'accès de base du projet, les fichiers graphiques, les tables de
langues, les tables de rapports et les fichiers média sont toujours transférés.
Les éléments suivants sont transférés depuis le chemin d'accès de base, par défaut : Les fichiers
project.ini, **Projekt.vba**, **Monitor.mon** et le dossier **Projekt**.

Par défaut, zenon utilise systématiquement des chemins relatifs (et non des chemins absolus),
afin que les fichiers puissent facilement être retrouvés sur le système cible.

Pour les fichiers dont le transfert est facultatif, vous devez toujours utiliser les chemins originaux
(la zone de texte vide dans "Cible"), de manière à ce que zenon puisse les trouver sur le système
cible.

PROJET GLOBAL

Si un projet global est chargé dans l'espace de travail, celui-ci est automatiquement transféré. Vous n'avez besoin de définir aucun paramètre supplémentaire. Tous les fichiers nécessaires pour le projet global sont toujours transférés.

> ⚠ **Attention**
>
> *If the time difference between the server and the client is more than 5 seconds, no more files are synchronized.*

14.6   Extended Trend

Extended Trend shows information from archives and online data. This data is saved in the Primary Server and requested by the Primary Server if required (if a trend screen is called up on the client).

14.7   Fonctions

Pour les fonctions utilisées sur le réseau :

▸ L'emplacement d'exécution peut être librement configuré, dans certains cas

▸ L'emplacement d'exécution est prédéfini, dans d'autres cas

> 💡 **Informations**
>
> *Les scripts associent plusieurs fonctions. L'emplacement d'exécution dépend alors des paramètres de la fonction **Exécuter le script**. Ce paramètre remplace les paramètres des fonctions individuelles.*

CONFIGUREZ L'EMPLACEMENT D'EXÉCUTION

Pour les fonctions pour lesquelles l'emplacement d'exécution peut être librement configuré, les paramètres correspondants sont disponibles dans les propriétés de la fonction.

Pour définir l'emplacement d'exécution :

1. Accédez au groupe **Exécution** dans les propriétés du projet.

2. Sélectionnez l'emplacement d'exécution souhaité en cochant la case. Une sélection multiple est possible :

   • **Ordinateur actuel** : La fonction sera exécutée sur l'ordinateur actuel.

- **Serveur principal** : La fonction sera exécutée sur le serveur principal.

- **Serveur redondant** : La fonction sera exécutée sur le serveur redondant.

- **Client** : La fonction sera exécutée sur la totalité des clients.

VUE D'ENSEMBLE DES FONCTIONS DE RÉSEAU

La table suivante illustre les fonctions exécutées, ainsi que leur emplacement d'exécution.

Légende :

▶ **Paramétrable :** le comportement peut être configuré

+: Oui

-: Non

O : Par défaut

▶ Si non paramétrable, O identifie l'emplacement d'exécution :

- Ordinateur actif

- Serveur primaire

- Serveur redondant

- Client

Fonction	Paramétrable	Ordinateur actuel	Serveur principal	Serveur redondant	Client
Alarmes et événements					
Alarmes : acquitter clignotement	-	O			
Alarmes : supprimer	-		O	O	
Acquitter les alarmes	-		O	O	
Inscrire/désinscrire groupe d'alarmes/d'événements	-	O			
Activer/désactiver liste d'alarmes / groupe d'alarmes/d'événements / classe d'alarmes/d'événements	-		O	O	
Gestion d'alarmes active	-		O		
Liste d'alarmes active/inactive	-		O		
Liste d'alarmes inactives	-		O		
Exporter liste d'alarmes	+	O			
Enregistrer le buffer des alarmes et des événements	-		O	O	
Exporter CEL	+	O			
Imprimer liste d'alarmes ou d'événements	+	O			
Créer/imprimer un document IPA	-		O		
impression en ligne on/off	-		O	O	
Démarrer nouvelle page pour l'impression en ligne	+	O			
Changer imprimante en ligne	-		O		

Application					
Sélection imprimante	+	O			
Démarrer la Gestion de charge	-		O		
Arrêter la Gestion de charge	-		O		
Imprimer le diagramme Extended Trend	+	O			
Changer de palette	+	O			
Fonctions : activer sur valeur limite	-		O	O	
Fonctions : activer/désactiver sur valeur limite	-		O	O	
Fonctions : désactiver sur valeur limite	-		O	O	
Ouvrir l'aide	+	O			
Réinitialiser projet en ligne	+	O			
Détermination des maintenances ouvertes	-		O		
PFS – Exécuter un événement défini par l'utilisateur	+	O			
Activer/désactiver la simulation de projet	-	O			
Simuler le clic droit	+	O			
Enregistrer les données rémanentes	+	O			
Arrêter le Runtime	+	O			
Analyser les heuristiques du graphe S7	+	O			
Exécuter une fonction SAP	+	O			

Changement de langue	+	O			
Topologie - Recherche des défauts de terre	-		O		
Topologie – Délestage	-		O		
Serveur d'archives					
Archive : Arrêter	-		O	O	
Indexer les archives	-		O		
Archive : Début	-		O	O	
Exporter les archives	-	O			
Archive : liste des archives actives	-		O	O	
Gestion des utilisateurs					
Changer d'utilisateur	+	O			
Connexion avec boite de dialogue	+	O			
Connexion sans mot de passe	+	O			
Déconnexion	+	O			
Modifier mot de passe	-	O			
Synoptiques					
Modifier la couleur source de la fonction CAL	+	O			
Synoptique indexé	-	O			
Synoptique : fermeture	+	O			
Synoptique : retour au précédent	-	O			
Synoptique : déplacer le centre	+	O			
Appel de synoptique	+	O			
Activer l'élément qui a le focus	+	O			
Donner le focus au cadre	+	O			
Déplacer le focus	-	O			
Enlever le focus du cadre	+	O			
Visualiser menu	+	O			
Sélection moniteur	+	O			
Profils de Runtime	+	O			
Fermer le cadre	+	O			

Saisie de valeur pour synoptique clavier	-	O			
Affichage de la fenêtre d'aperçu	+	O			
Localisation de défauts sur les réseaux électriques					
Acquitter le message de défaut de terre	+	O			
Arrêter la recherche des défauts de terre	+	O			
Démarrer la recherche des défauts de terre	+	O			
Acquitter le message de court-circuit	+	O			
Message Control					
Enregistrer queue courante	-		O		
Supprimer des groupes, classes, zones ou équipements	-		O		
Envoi d'un message	-		O		
Envoi de messages : activer	-		O		
Envoi de messages : désactiver	-		O		
Réseau					
Autorisation sur le réseau	+	O			
Commutation redondance	-			O	
Report Generator					
Imprimer rapport	+				
Rapport : exécuter	+				
Exporter rapport	+				
Recettes					
Recipe Group Manager	-	O			
Recette standard	-	O			
Recette : envoi direct	+	O			
Recette : modification avant envoi	-	O			
Recette : sélection et modification avant envoi	-	O			

Script					
Script : exécuter	+	O			
Script : sélection en ligne	+	O			
Variable					
Exporter données	-		O		
Lire un fichier dBase	+	O			
Imprimer valeurs actuelles	+	O			
Unit conversion (Conversion d'unités)	+	O			
Gestion HD active	-		O	O	
Gestion HD inactive	-		O	O	
Gestion HD inactive/active	-		O	O	
Écrire valeur prescrite	-		O		
Commandes du driver	-	O			
Transférer l'image de simulation du driver vers le serveur de secours	-			O	
Écrire l'heure dans la variable	+	O			
Lire l'heure depuis la variable	+	O			
VBA					
Ouvrir l'éditeur PCE	-		O		
Ouvrir l'éditeur VBA	+	O			
Exécuter macro VBA	+	O			
Montrer la boîte de dialogue des macros VBA	+	O			
VSTA					
Ouvrir l'éditeur VSTA	+	O			
Exécuter macro VSTA	+	O			
Montrer la boîte de dialogue des macros VSTA	+	O			
Windows					
Lire un fichier audio	+	O			
Opérations sur fichiers	+	O			
Son bip : début	+	O			

Son bip : fin	+	O			
Fenêtre en arrière-plan	-	O			
Fenêtre en avant-plan	-	O			
Imprimer la capture d'écran	+	O			
Exécuter un programme	+	O			

14.8   Message Control

Message Control will be executed on the Primary Server.

The Primary Server synchronizes with the Standby Server.

14.9   Programming interfaces

VBA AND VSTA

Code in VBA or VSTA is always, by default, executed locally on the system on which it is started or where events occur.

The place of execution can however be defined otherwise when this is called up via the function (à la page 136).

PCE

PCE is always executed on the server in the network. On standalone computers in standalone projects.

14.10   Cross-reference list

The use of variables in rated networks is also taken into account. The property name of the place of usage is stated as the element name. This is either "event variable" or "evaluations".

<table>
<tr><td>💡 Informations</td></tr>
</table>

You can find out further information in the cross-reference list manual.

14.11 Report Generator & Report Viewer

The ***.xrs** files of the Report Generator and the ***.rdl** files of the Report Viewer are synchronized on all systems in the network (Clients, Standby Server, Primary Server).

EDITOR

If the file is modified in the zenon Editor, transferred to the Primary Server and reloaded, it is automatically distributed to the other computers in the network.

RUNTIME

If the file is amended in Runtime, the changes are only saved on a temporary basis and replaced the next time a reload takes place or when Runtime is restarted.

14.12 Recipes

The execution of recipes is different for standard recipes and the RGM.

STANDARD RECIPES

Standard recipes are managed on the Primary Server and on the Standby Server.

If a standard recipe is changed by a user in Runtime, the client requests the full recipe list from the Primary Server. In the event of changes, the recipe list is sent back to the Primary Server.

<table>
<tr><td>💡 Informations
This list is not identical to the file rezepturen.cmp</td></tr>
</table>

If a recipe is changed and executed in Runtime on the client, it is executed with the new values. When the standard recipe, you are given the option to save the changes.

RECIPEGROUP MANAGER

When the Recipegroup Manager screen is loaded on the client, a list of all recipe names is requested by the Primary Server. As soon as a recipe is selected, it is loaded by the Primary Server.

14.13 Command Sequencer

The network concept of the **Command Sequencer module** works according to the following principle:

▶ The command sequences can be configured on the client, as well as the **Server** or on the **Standby Server**.

▶ The configured command sequences are administered on the **Primary Server** and distributed to the clients.

▶ The command sequences can be operated on both the client and the **Server**.

▶ The command sequences are always executed on the **Primary Server**.

▶ With redundancy switching, the command sequences are canceled.
These can be restarted manually on the new **Primary Server**.

ROLE SWITCH BETWEEN SERVER 1 AND SERVER 2

▶ Redundancy switching is delayed until all active command sequences have been completed.

▶ The start of command sequences is blocked during a redundancy switching. The buttons on the client are grayed out in this time.

- This redundancy switching can be planned in a **Rated network**.

- In a **dominant network** or a **non-dominant network**, redundancy switching is carried out when the **Primary Server** fails.

▶ It is possible to start command sequences again once the switch has been carried out or if the switch has been completed.
An entry is written to the CEL in this case.

▶ CEL messages are written for the following events:

- Start of a command sequence on the server is blocked.

- If a command sequence in the dominant network is to be started on the **Serveur 2** .

- The command sequence cannot be started because there is currently a redundancy switch pending.

- Note: No CEL message is generated if an incorrect command sequence is started.

LOG ENTRY

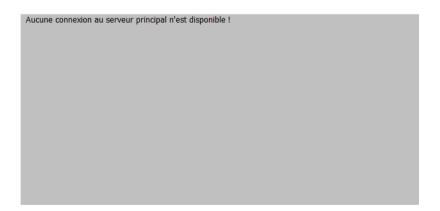| Entry	Level	Description
The sequence (mrid:<id1>, crid:<id2>)<name> could not be started, because a redundancy switch is pending.	ERRORS	The command sequence cannot be started because there is a redundancy switch pending.

SPECIAL CASE: TWO SERVERS IN THE NETWORK

In the event that, when switching the **Primary Server**, there are still command sequences running on the "new" **Standby Server**, these are canceled on the Standby. This can only occur if both servers in the network were no longer connected (due to a network failure for example) and are now connected again. In this case, the change to the command sequence is not transferred to the Primary Server.

This means that, if there is a connection and command sequences that are now canceled on the Standby have already been opened on the Primary Server, these continue to be considered as running. It can only be restarted again once this command sequence has been closed on **Serveur 1** and **Serveur 2**.

NO CONNECTION TO SERVER AND STANDBY

If the command sequence screen is opened on the Client when neither **Serveur 1** nor **Serveur 2** are contactable, the command sequence editor on the Client is not available. The command sequence editor remains empty. An error text is displayed in zenon Runtime instead of the command sequence image.

Aucune connexion au serveur principal n'est disponible !

ERROR DIALOG

If a command sequence cannot be started, the following error dialog is called up:

14.14 Scripts

Scripts combine several functions. The place of execution depends on the settings of the **Script : exécuter** function. This setting overwrites the settings of the individual functions.

The execution of scripts in the network is controlled with predefined scripts:

Script	Description	Place of execution
AUTOSTART	The script is executed automatically when Runtime starts before the start screen is loaded if the project is the Runtime start project. It is not executed when subordinate projects are started.	▶ **Network project:** Primary Server ▶ **Single-user project**: Active computer
AUTOEND	The script is executed automatically when Runtime is ended if the project is the Runtime start project. It is not executed when subordinate projects are ended.	▶ **Network project:** Primary Server ▶ **Single-user project**: Active computer
AUTOSTART_CLIENT	The script is executed automatically on a client when Runtime starts before the start screen is loaded if the project is the Runtime start project. It is not executed when subordinate projects are started.	**Single-user project**: Active computer
AUTOEND_CLIENT	The script is executed automatically on a client when Runtime is ended if the project is the Runtime start project. It is not executed when subordinate projects are ended.	**Single-user project**: Active computer
AUTOSTART_SRVPRJ	Script is executed automatically when Runtime is started for any desired project on the Primary Server before the start screen is loaded.	**Network project:** Primary Server
AUTOEND_SRVPRJ	Script is automatically executed when Runtime of a desired project is ended on the Primary Server.	**Network project:** Primary Server

14.15 Drivers in the zenon network

In the zenon network, the Primary Server normally communicates with the controller via the driver. Requests from the Client are routed via the Primary Server. This gets the information from the controller and forwards it to the Client. Limit values are monitored by the Primary Server.

DRIVERS

Drivers are executed on the Primary Server and Standby Server.

14.15.1 Driver for internal variable

With internal variables, it is possible in zenon to stipulate, for each variable, whether the variable is local on the computer or the same in the complete network.
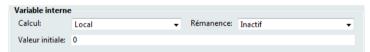
> 💡 **Informations**
>
> **Driver object type** `internal variables` are only available for the **driver for internal variables(INTERN)**.

To start calculating internal variables:

1. Navigate to the **Variable interne** node in the properties of internal variables

Variable interne			
Calcul:	Local	Rémanence:	Inactif
Valeur initiale:	0		

2. Stipulate the place of execution using the **Calcul** property:
    This property can be set individually for each variable. Arrays are also supported.

    a) `Local`: The internal variable is evaluated or managed locally for network projects, i.e. also on the Standby Server and the Clients. The values are not synchronized with other computers in the network.

    b) `Network`: With network projects, the internal variable is evaluated or administered on the project's Primary Server. It has the same value on the Primary Server and all Clients.

Remarque : Indisponible si le terminal CE est utilisé en tant que serveur de données. D'autres informations sont disponibles dans le manuel zenon Operator, au chapitre Terminal CE en tant que serveur de données.

14.15.2 Drivers - Limit values and reaction matrices

In principle, limit values and reaction matrices are monitored on the Primary Server (where the AML is also administered).

The following applies for local internal variables in addition:

▶ Limit values and reaction matrices are monitored locally on all computers (Primary Server, Standby Server and Clients).

▶ Linked limit values are only executed where the limit value of the local internal variable was violated.

▶ Limit value violations of these variables on the Standby Server or Client are not an alarm.

14.15.3   Driver - regrading delay in the event of redundancy switching

When a configured server is started or restarted, the regrading to the Primary Server can be delayed by a driver.

This ensures that regrading only takes place if all process variables have a valid value.

> 💡 **Informations**
>
> The regarding delay is only supported by very few drivers, such as the Siemens AK-Treiber for example
>
> You can find out whether a regrading delay is supported in the respective documentation of the driver.

LOG MESSAGES

Parameters	Level	Description
Redundancy Switch confirmation requested for project <Project name> module:VAR sequenceNo:<SequenceNo>	**Debug**	Confirmation that the Primary Server can be switched requested ▶ `Project name` The project name ▶ `SequenceNo` Serial number for the assignment of `request` and `response`.
Redundancy Switch confirmed for project <Project name> module:VAR sequenceNo: <SequenceNo>	**Debug**	Confirmation that the Primary Server can be switched.
Redundancy Switch delayed for project <Project name> module:VAR sequence id: <SequenceNo>	**Debug**	Delay activated for regrading.
Sample driver gave startup okay	**Debug**	Delay time on the driver expired.
me:<transport class> mod:<NetzwerkModuleId>(<network module name>) msg:<network module command> SeqId:<SequenceId> dest:<Target computer> prj:<project name>	**Debug**	LOG entry if cross-computer confirmation is necessary: ▶ `Transport class:` Class name for the transport of the command. Varies according to module. ▶ `Network module ID:` ID of the network module. ▶ `Network module name:` Name of the network module. ▶ `SequenceId:` Serial number that is used to assign `request` and `response`. ▶ `Target computer:` Computer on which the data is transferred. Optional. `S` denotes the active server ▶ `Project name:` Name of the project for which the transport is executed.

14.16 zenon web client in the redundant network

With a redundant zenon network configuration, the issuing of roles for the primary server and standby server depends on the redundancy type. It is not always clearly recognized whether the configured **Serveur 1** or**Serveur 2** is in the role of the Primary Server.

You should therefore configure both servers for the web server. You configure this configuration in global_vars.js with the **RUNTIMESERVER=.** entry

In doing so, the sequence should conform to the configuration in the zenon Editor:

- ▶ `Runtime server ` = **Serveur 1**
- ▶ `Runtime server2` = **Serveur 2**

The zenon thus attempts to contact both servers after one another. If, for example, the configured `Runtime server2` is the Primary Server, it sends the data (including the project.ini file) to the client.

AMENDED SERVER CONFIGURATION

If the server names configured in the Editor do not correspond to the server names of **global_vars.js**, the web client or the client Runtime will not start.

If the configuration of the server is amended for a running system in zenon, the "`Runtime is busy` dialog will be shown in the zenon web client.

After a project synchronization, the currently-running and the actual project configuration will be shown in a further dialog. In this case, the browser window must be closed by the user and the web client must be restarted.

> ⚠ **Attention**
>
> *If the project configuration of **Serveur 1** and**Serveur 2** is amended in the zenon Editor, the `global_vars.js` file must also be amended accordingly.*
>
> *You can find further information in the zenon web server manual in the configuration of global_vars.js chapter.*

14.17 Time Control

Time control will be executed on the Primary Server. The function triggered is executed on the systems that were selected for execution of the function in the settings.

14.18   Allocations

Allocations are always executed on the Primary Server.

> ⚠ **Attention**
>
> This is relevant to local internal variables most of all. These are not executed on the Standby Server or on Clients!

15. Network messages from the system driver

The following system driver variables are available for this subject area:

Name	Data type	Comment

[Network] Current Primary Server	`STRING`
	
	`lokal`
		Computer name of the current Primary Server
		If the name was acquired from the host file, it will be the name used there. With DNS, this is the Fully Qualified Domain Name.
		
		Note: If the network is deactivated, the variable sends the status INVALID. The **[Network] Current Standby Server** variable remains empty in contrast.

[Network] Current Standby Server	`STRING`
	
	`lokal`
		Computer name of the server which is currently not handling processes.
		If the name was acquired from the host file, this is the name entered there. With DNS, this is the Fully Qualified Domain Name.

[Network] Number of connected clients	`UDINT`	Delivers the number of clients currently connected to the server. This number also includes the standby server, if there is one.
[Network] Authorization denied	`BOOL`	Shows whether a requested authorization is denied in the network. The value of this variable is changed only for a short time and then changed back to the initial state. ▸ `0` = operating authorization request accepted ▸ `1` = operating authorization request declined
[Network] Authorization available	`BOOL`	Shows whether there is an authorization for the current project on the local computer. ▸ `0` = No ▸ `1` = Yes
[Network] operating authorization: Computer that has it	`STRING`	Shows the name of the computer that has the authorization for the currently loaded project.
[Network] Evaluation result of Server 1	`UDINT`	In the event of changes to a variable from the evaluation matrix, this value is written to the corresponding system driver variable for Server 1 and Server 2 after calculation of the new result of the evaluation. The values are equal to one another (server <-> standby), so that the current value is always displayed on both sides. However, after the other side has a failure, this value remains for the attendant variable and only updates itself once it reconnects. Note: You can find further information on evaluation in the Network (à la page 7) manual in the Configuration of redundancy evaluation (à la page 84) chapter.

[Network] Evaluation result of Server 2	`UDINT`	In the event of changes to a variable from the evaluation matrix, this value is written to the corresponding system driver variable for Server 1 and Server 2 after calculation of the new result of the evaluation. The values are equal to one another (server <-> standby), so that the current value is always displayed on both sides. However, after the other side has a failure, this value remains for the attendant variable and only updates itself once it reconnects. Note: You can find further information on evaluation in the Network (à la page 7) manual in the Configuration of redundancy evaluation (à la page 84) chapter.
[Network] Names of connected clients	`STRING`	xxx Delivers the names of the clients currently connected to the server. The standby server, if there is one, is also included.
[Network] Primary Server <-> Standby Server in data sync	`BOOL`	A binary variable that takes the value `1` (for a short time) when the system performs a redundancy switch between server and standby server. ▸ `0 = no file sync` ▸ `1 = file sync active`

[Network] Primary Server broke down	`BOOL`
	
	`lokal`
		Indicates that the connection to the process handling server was lost.
		Depending on the network position of the computer, this means:
		
		▸ Dominant Server: While it is not yet the process handling server, the value changes to TRUE if the connection to the process handling server is lost. Always FALSE after synchronization.
		
		▸ Non-dominant Server: Changes to TRUE if the connection to the dominant server, which was the process handling server, is lost. Changes back to FALSE if the Standby Server was promoted to be the Primary Server.
		EVALUATION: Preferably via a reaction matrix (REMA), as the Alarm Management is also swapped and taken over by the SB at that time. The Online Container is also not suitable because the variables are re-initialized during redundancy switching.
		
		Client: Changes to TRUE if the connection to the process handling server is lost. Changes back to FALSE if the client connects to the SB computer that is now the process handling server.

[Network] Primary Server shut down	`BOOL`	Indicates the regular stop of the process handling server. The value changes to TRUE if the Primary Server was stopped

	`lokal`	properly. FALSE if there is a process handling server in the net. Depending on the network position of the computer, this means: ▸ Dominant Server: While it is not yet the process handling server, the value changes to TRUE if the process handling server has stopped. ▸ Non-dominant Server: Changes to TRUE if the dominant server, which was the process handling server, has stopped. Changes back to FALSE if the StandBy was promoted to be the process handling server. ▸ EVALUATION: Preferably via a reaction matrix (REMA), as the Alarm Management is also swapped and taken over by the SB at that time. The Online Container is also not suitable because the variables are re-initialized during redundancy switching. ▸ Client: Changes to TRUE if the dominant server has stopped. Changes back to FALSE if the client connects to the SB computer that is now the process handling server. Is also TRUE while the process handling non-dominant server changes back to be the non-process handling server.
[Network] Standalone/Primary Server/Standby Server/Client	`DINT`	Shows the type of the local computer in the network. ▸ `−1` = Standalone ▸ `0` = Client ▸ `1` = Primary Server ▸ `2` = Standby Server
[Network] Standby Server broke down	`BOOL`	Changes to TRUE if the connection to the currently non-process handling server is terminated unexpectedly. If there is a connection, the value is FALSE. Depending on the network position of the computer, this means: ▸ Dominant Server: The variable only acts as described from the time when the standby became the server handling the process. ▸ Non-dominant Server: If, during file synchronization, the connection to a server that is dominant but is not handling the process is interrupted, the value changes to TRUE. Always FALSE if not the server handling the process. ▸ Client: As per server handling the process.

[Network] Standby Server shut down	BOOL	Is TRUE on the process handling server, if the non-process handling server was stopped properly and if there is no connection anymore. Changes to FALSE if the non-process handling server has registered at the process handling server. Depending on the network position of the computer, this means: ▸ Dominant Server: Only from the time when the standby became the server handling the process does the variable act as described. ▸ Non-dominant Server: If this is ended during file synchronization with a server that is dominant but is not handling the process, the value changes to TRUE. Always FALSE if not the server handling the process. ▸ Client: As per server handling the process.
[Network] Standby Server started	BOOL	Is TRUE if the non-Primary Server has signed into the Primary Server, the file sync was carried out and the connection between the two computers is active. Depending on the network position of the computer, this means: ▸ Dominant Server: Only from the time when the standby became the server handling the process does the variable act as described. ▸ Non-dominant Server: Becomes TRUE if the dominant server not handling the process starts. Changes to FALSE if the computer is the server handling the process. ▸ Client: As per server handling the process.
[Network] Timeout [ms]	UDINT	Shows the timeout in milliseconds for the zenon network as configured in the project configuration.
[Network] Switch from Primary Server to Standby Server	BOOL	A binary variable that takes on the value 1 if the server becomes the standby server during a redundancy switch. ▸ 0= registered server is available as server in the network. ▸ 1 = registered server is available as standby server in the network.
[Network] Switch from Standby Server to Primary Server	BOOL	A binary variable that takes on the value 1 if the standby server becomes the server during a redundancy switch. ▸ 0 = registered Standby Server is available as standby server in the network. ▸ 1 = registered Standby Server is available as server in the network.